CS10454

Create Complex Building Site Tools with Revit Families and API

Merri LAWAN
VINCI CONSTRUCTION FRANCE

Learning Objectives

- Understand certain family functionalities integrating geometry, data and logic rules
- Envision how API can help overcome the limitation in families by simplifying their creation and improve their overall use
- Envision how API can help overcome the limitation in families by simplifying their data acquisition and synchronization
- Understand how the API can help organized business logic by creating more suitable user interfaces

Description

This class will discuss how to create complex building-site parametric tools (crane, formwork, security) thanks to Revit software family's technologies and its API. We will focus on the practical-use cases where the combination of these 2 technologies can assist engineers in the proper choice of crane, optimized formwork, scaffolding, security materials, and other site equipments. In order to achieve this, it is crucial to understand both the manufacturers and our company's security and productivity rules and guidelines and synchronize them both to best equip our engineers and sites.

Your AU Experts

With over a decade of experience in construction both on site and off site, ***Merri*** *joined the French leading general contracting company Vinci Construction France's BIM team at its inception in 2011. Having worked for Bouygues Construction,* ***Merri*** *has been a keen Revit user since 2010 and as of 2011, He is now responsible for the BIM implementation in Vinci's Construction France, in charge of Revit integration within design teams, technical departments and cost estimations to construction sites.*

As an Autodesk Developer Network Member (ADN), ***Merri*** *focuses on custom software development and takes an active part in the Autodesk Beta Testing Program. As an expert in the Revit software API, he has developed and deployed Vinci's in-house plugins and tools accelerating business workflow among BIM products. Finally, as a BIM evangelist in Vinci, his role includes the training supervision and technical support for all subsidiaries in France.*

Table des matières

1 Introduction

The goal of this presentation is to show you, through different use cases, how we start implementing a building site tool library in Autodesk Revit during the past few years. But before diving through the detail of this implementation in the software, I would like to explain quickly what is our main business, what lead us to create such intelligent tools and the actual and future benefits we focus on.

2 The context

2.1 The Group main activities and BIM knowledge

VINCI CONSTRUCTION FRANCE (VCF) is the French subsidiary of Vinci Group (World leader Construction Company). We are 20 000 employees spread in a lot of small business units in France. VCF is a general contractor, and his main activity is the concrete construction. That means that Vinci's engineers are specialized in concrete staff and other jobs are subcontracted but Vinci deal with the overall coordination and logistics.

There are 4 main jobs in VCF:

- Cost estimators realize the overall pricing study and planning
- Temporary worksite designers and site planners also called "Methods": I will come back to this later because they are the only parties concerned by the building site tool library in Revit. They also play a key role in the company process
- Structural Engineers realize structural load calculations, rebar and concrete dimensioning, as well as execution plans
- Building site engineers coordinate financial and human resources, and also ensure the overall coordination and security.

VCF started to implement BIM in 2011. The subject became more strategic since the European Council decided last year to make the BIM mandatory in all public tender.

In Paris, a team of ten people has first been created to answer different questions:

- Which BIM software should Vinci use?
- What is the gain of BIM inside the company?
- How to structure the organization around BIM Projects?

Now the team role is more about:

- Implementing templates, libraries and plugins to speed up document production
- Training

2.2 The Methods job and his key role inside Vinci

Historically, Methods engineers were the first who tried to embrace the BIM, because they need the 3D so as to resolve and explain specific building construction operating procedures, show safety and shoring layout plans.

At the same time, since 10 years, VINCI CONSTRUCTION tend to homogenize the way of preparing and organize building sites into one single common process called Orchestra, by picking the best practices from the subsidiaries.

As we can see below, the process look like a wheel (like BIM) and Methods Engineers are playing the biggest part of it (blue and green section of the Orchestra wheel).

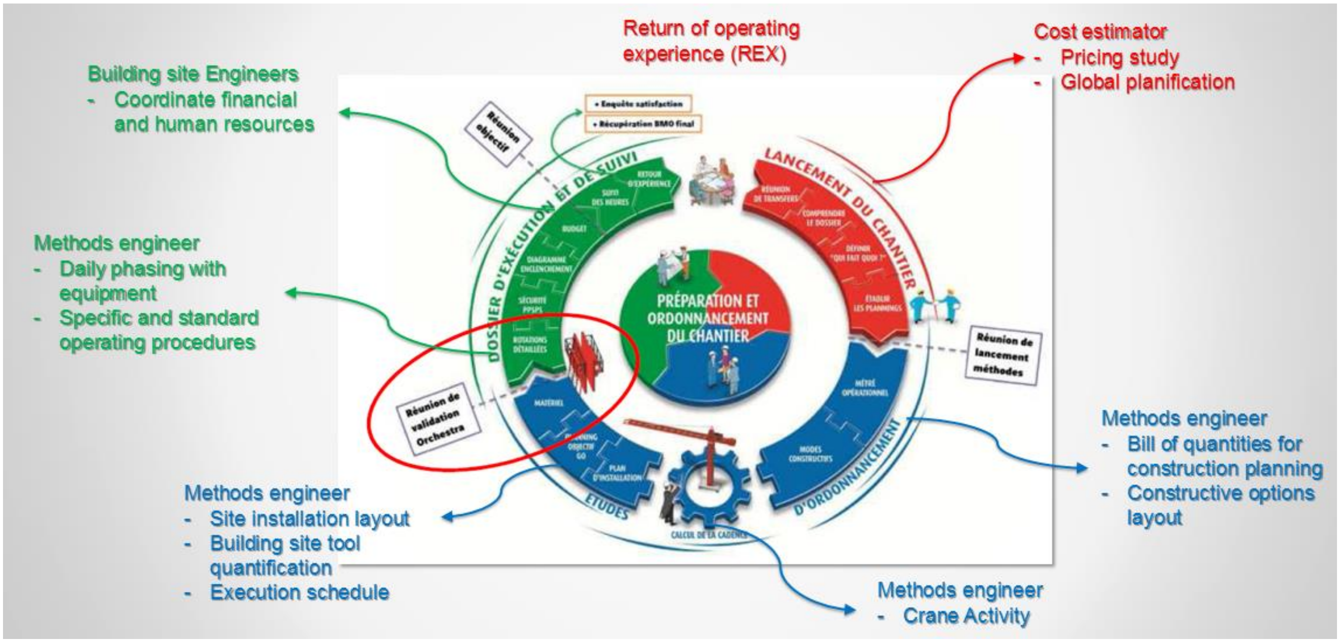
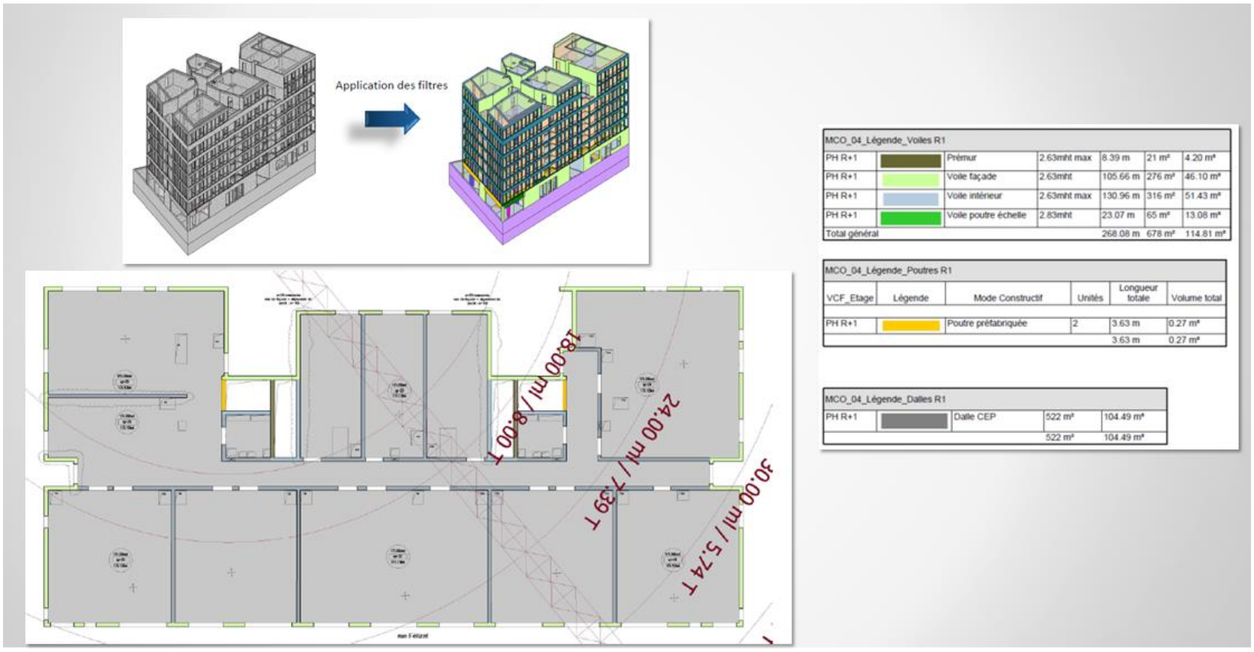


FIGURE 1: ORCHESTRA PROCESS

According to the Orchestra process, Methods production documents are the following:

- Bill of quantities to edit construction planning: the idea is to get automatically an Excel document which describes quantities per elements (wall, column, beam ….) and areas, and group these quantities according to specific dimensions and construction methods (wall cutting by formwork available heights for example).

- Constructive options layout: this consists on a project per-element color visualization according to the construction methods chosen (different color for concrete cast in place or precast for example)



MCO_04_Légende_Voiles R1						
PH R+1	■	Prémur	2.63mht max	8.39 m	21 m²	4.20 m³
PH R+1	🟩	Voile façade	2.63mht	105.66 m	276 m²	46.10 m³
PH R+1	🟦	Voile intérieur	2.63mht max	130.96 m	316 m²	51.43 m³
PH R+1	🟩	Voile poutre échelle	2.83mht	23.07 m	65 m²	13.08 m³
Total général				268.08 m	678 m²	114.81 m³

MCO_04_Légende_Poutres R1					
VCF_Etage	Légende	Mode Constructif	Unités	Longueur totale	Volume total
PH R+1	🟨	Poutre préfabriquée	2	3.63 m	0.27 m³
				3.63 m	0.27 m³

MCO_04_Légende_Dalles R1				
PH R+1	■	Dalle CEP	522 m²	104.49 m³
			522 m²	104.49 m³

FIGURE 2 EXAMPLE OF CONSTRUCTIVE OPTIONS LAYOUT DONE IN REVIT

- Crane dimensioning and positioning depending on the building footprint geometry, building components to lift and associated quantities, and site environment
- Crane activity calculation achieved by getting building liftable components quantities and applying crane duration for each task. Crane activity calculation is done to erect the planning.
- Site installation drawings
- Daily or weekly phasing with associated wall formwork, propping and safety equipment as well as material (concrete pouring volume or precast units) quantification. The most tedious part today is the daily block by block equipment layout because of the few intelligence of AutoCAD.

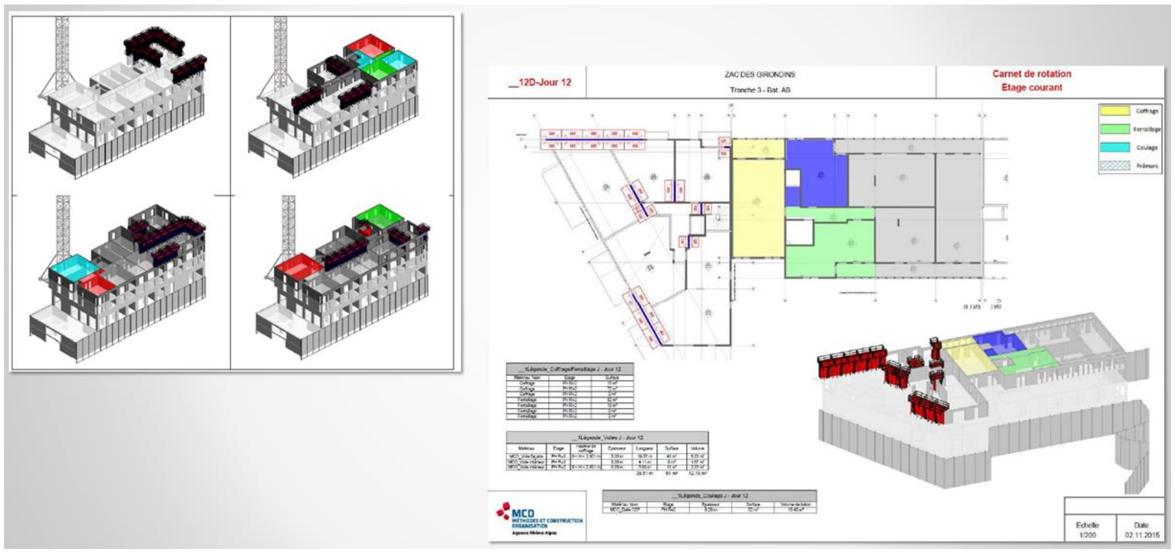


FIGURE 3 PLANNING DRAWING PER DAY WITH ASSOCIATED WALL FORMWORK

VCF was looking for a BIM software able to easily implement this process. Revit was chosen because of its following capabilities:

- Schedules for drawing quantifications
- Phase module and parts command in Revit for pour sequencing
- Color filtering by attributes to show construction relation of building components
- Equipment library creation: we can easily create highly parametric components in Revit using the family editor. For this specific point, everything has to be created from scratch.

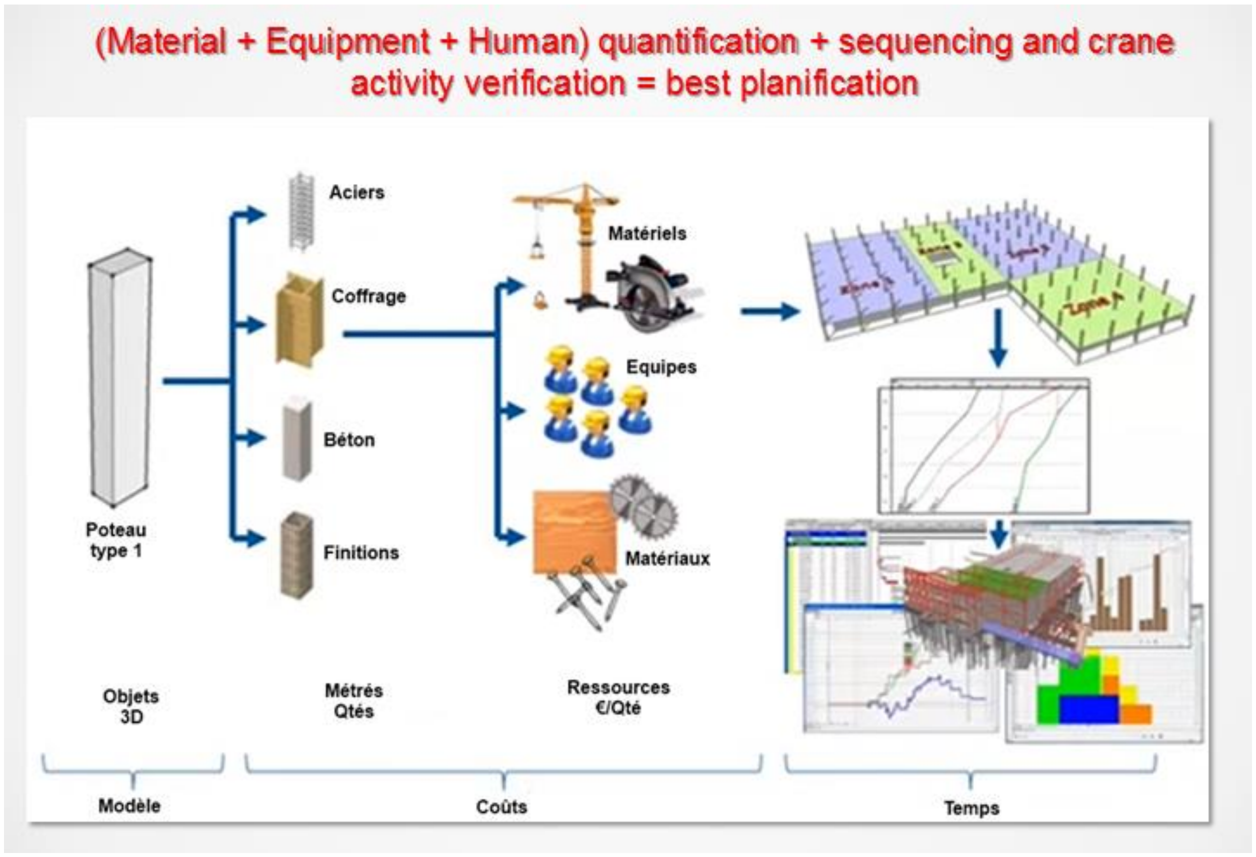


FIGURE 4 THE METHOD ENGINEER JOB

2.3 Why creating a building site tool library?

Methods Engineers are used to work with a huge 2D DWG static blocks library (very few dynamic blocks in fact) and they use a bunch of custom lisp / VBA tool in AutoCAD in order to speed up their drawings. For the quantification process, they use Excel data extractions (blocks attributes extractions) and introduce sequencing and construction methods notion by filtering with AutoCAD layers. Revit will immediately improve their workflow by adding 2 features:

- Scheduling by attributes or by components
- Sections and view plans are automatically updated

Most of the blocks in the Methods 2D library come from the building site tools providers who export 2D projections of their 3D fabrication models. Since it was 2D blocks without intelligence, it wasn't a big deal for manufacturers. But since the introduction of BIM, we started asking them for 3D models: most of them refused arguing it is too confidential. In some rare case we got it but it was not useable (too heavy detailed STEP files showing all the bolt).

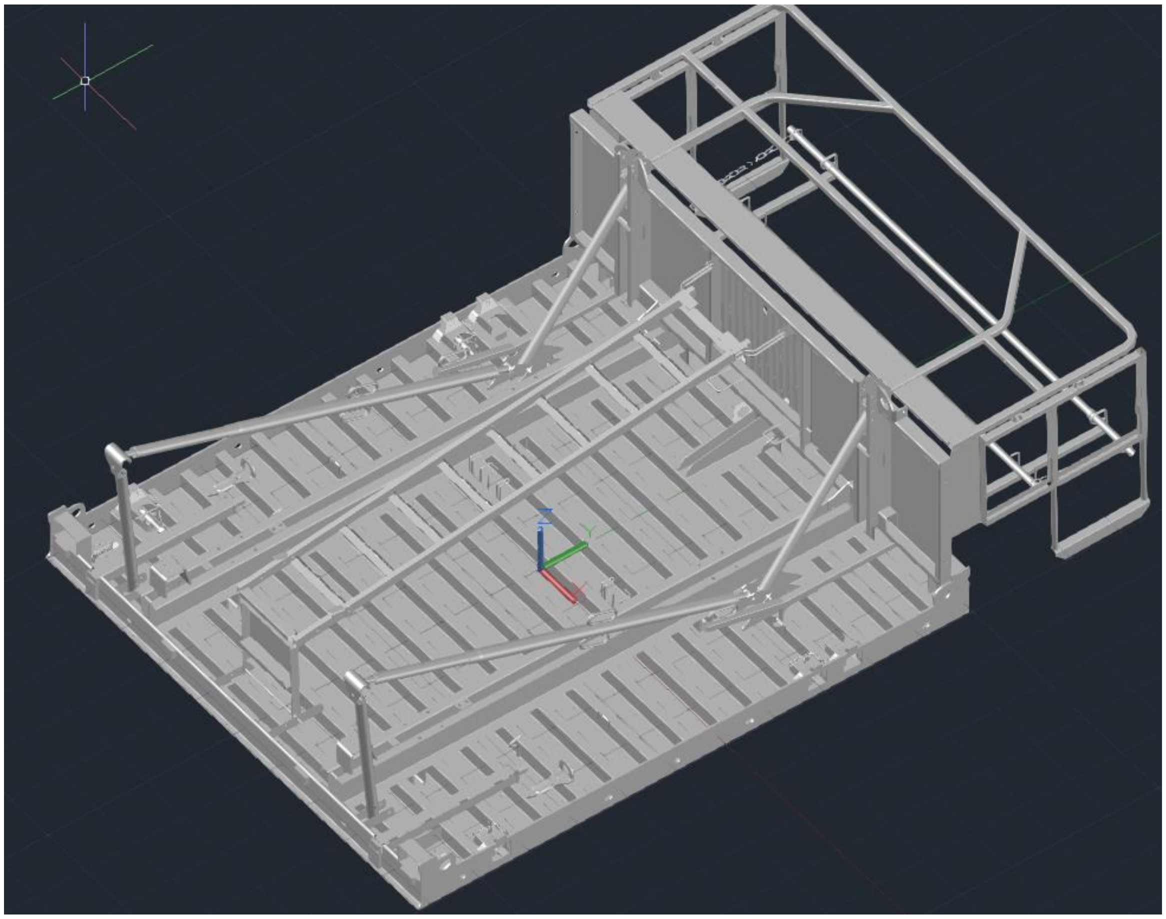


FIGURE 5 STEP FILE EXAMPLE FOR VERTICAL FORMWORK

The most advanced Methods offices in 3D were using Sketchup, because the 3D was easy and affordable, and because the Sketchup warehouse is the most popular platform where you can download 3D building site tools with a realistic aspect. But as it has been created by people and not the manufacturers, there is no guarantee about the dimensions (and it is really important when you make the temporary design drawing to deal with the exact space requirements). And last but not least, importing a Sketchup file in a Revit family slow down Revit models and graphics styles cannot be personalized (for example, apply Revit materials in order to render in Revit is not possible and you must stay in shaded view to see sketchup textures).

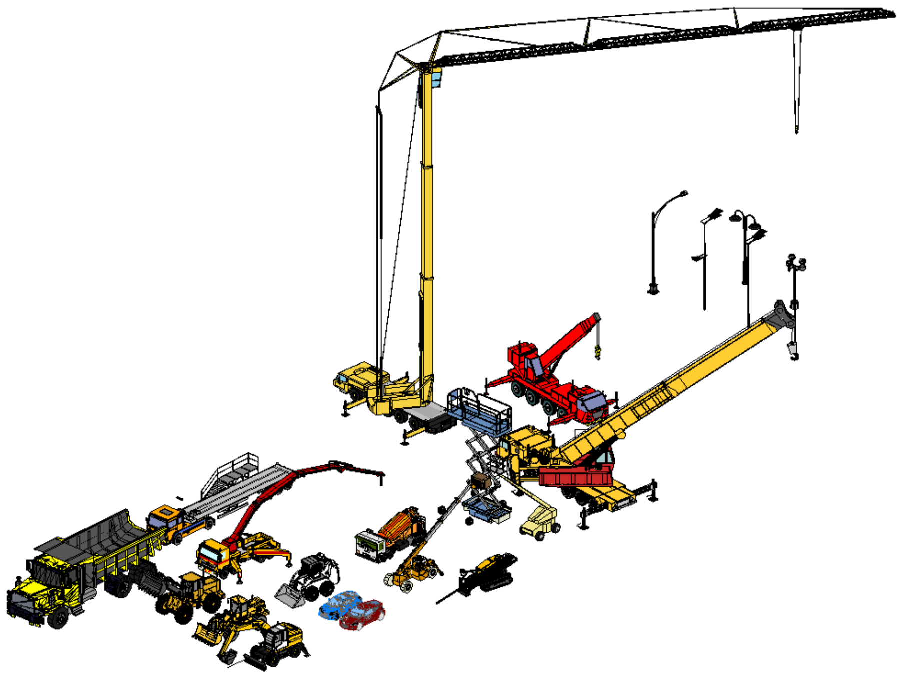


FIGURE 6 3D SKETCHUP EQUIPMENT IMPORTED INTO REVIT FOR BUILDING SITE INSTALLATION

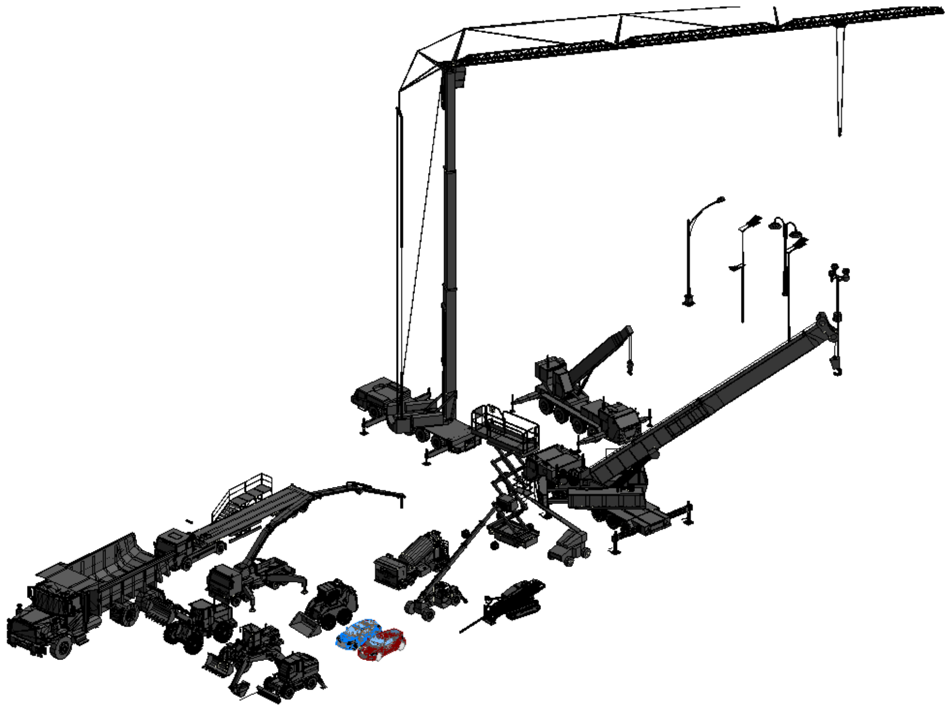


FIGURE 7 AND WHEN WE TRY TO RENDER IN REVIT... OUPS!

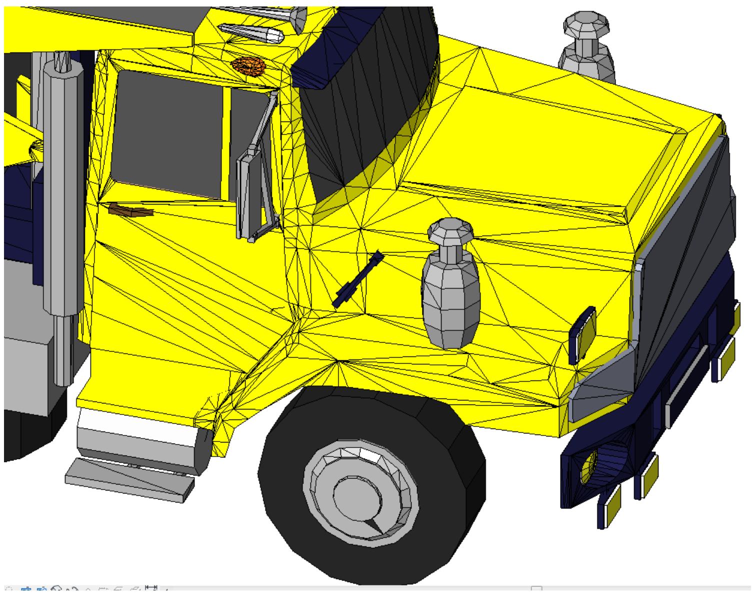


FIGURE 8 TRIANGULARISATION IN SKETCHUP SLOW DOWN REVIT PERFORMANCE!

2.4 Why Revit was chosen?

Here is an exhaustive list of the reasons Revit was chosen:

1 **Distinction between 3D views and 2D schematic views**
This is very important because in 2D, it is better to keep only the strict relevant details so that Engineers are able to label and make annotations without overloading the drawing

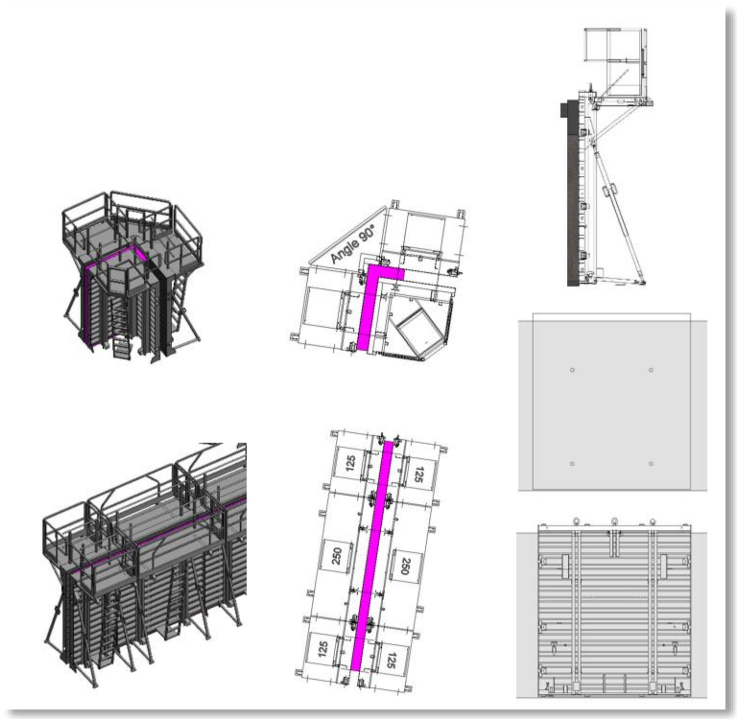


FIGURE 9 DISTINCTION BETWEEN 2D AND 3D VIEWS: WALL FORMWORK EXAMPLE

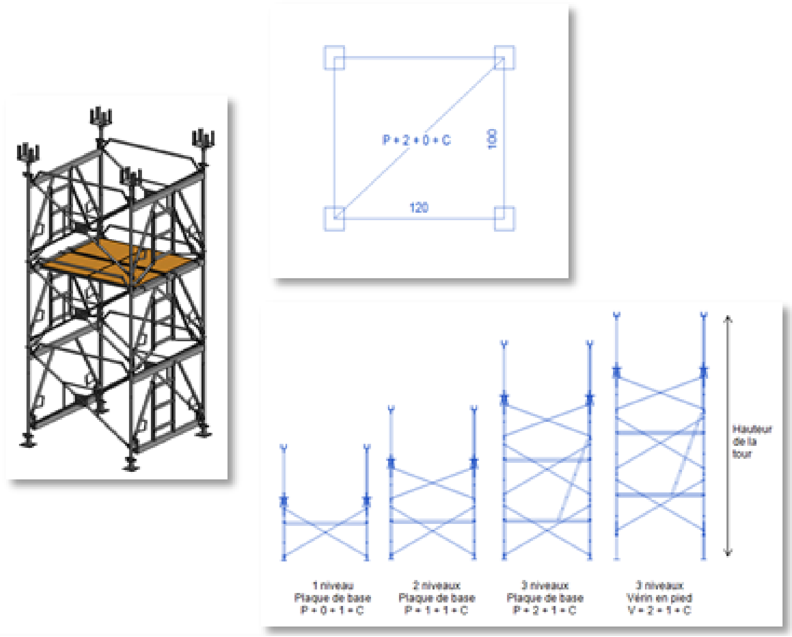


FIGURE 10 DISTINCTION BETWEEN 2D AND 3D VIEWS: SHORING TOWER EXAMPLE

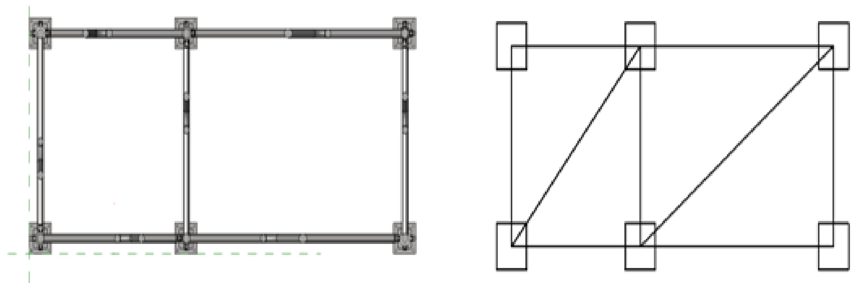


FIGURE 11 DIFFERENCE BETWEEN A 3D PROJECTED AND A 2D SCHEMATIC VIEW

2 Being able to start working with 2D components without being forced to model all the 3D details

This is a good transition from AutoCAD: Methods engineers can gradually migrate their library in 2D parametric block first (elevation and plan view) and then further model the 3D view when they have time. The advantage of working in 2D views in Revit compare to AutoCAD, is that 2D elevations and plan views are linked together and you can create parametric intelligence in 2D (similar to dynamic block modules in AutoCAD but a lot easier). Tower crane components is a good example because if you modify the jib, it will be modified both in 2D plan and elevation views which is not possible in AutoCAD (you still have to deal with two separated dynamic blocks).

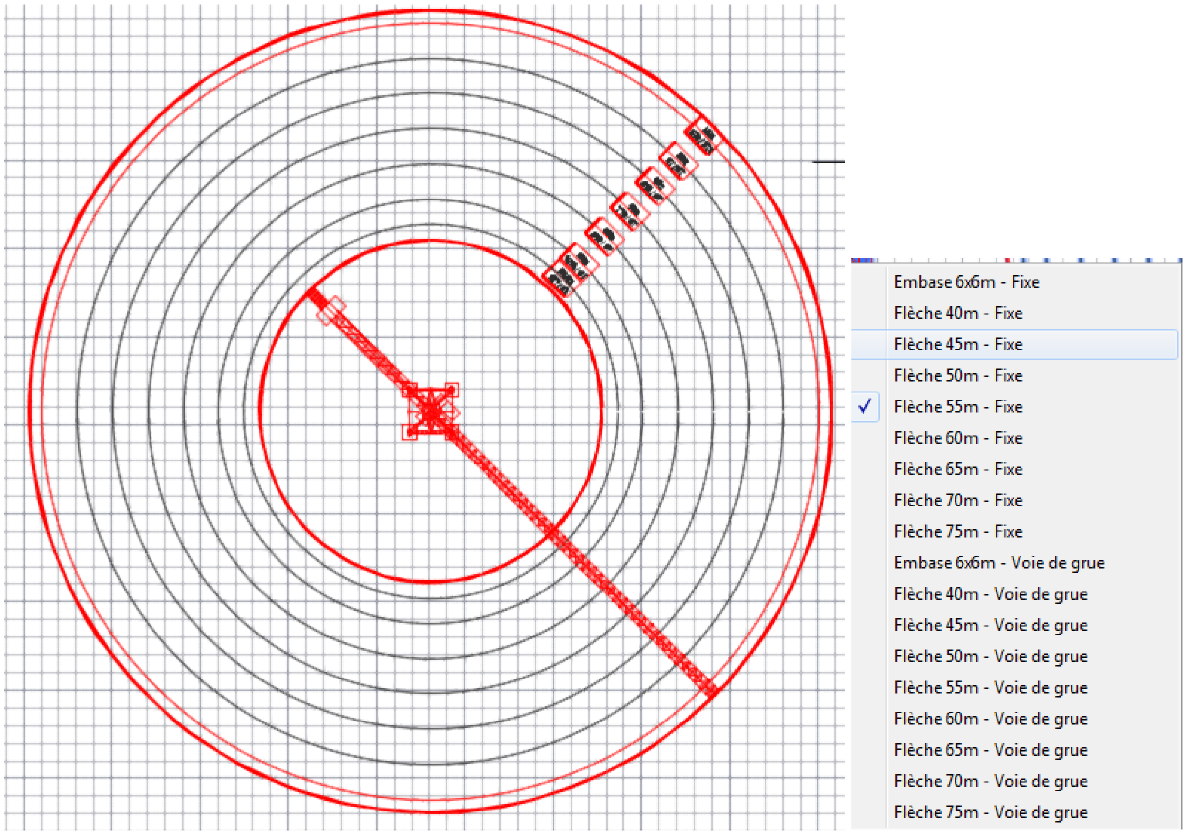

Embase 6x6m - Fixe
Flèche 40m - Fixe
Flèche 45m - Fixe
✓ Flèche 55m - Fixe
Flèche 50m - Fixe
Flèche 60m - Fixe
Flèche 65m - Fixe
Flèche 70m - Fixe
Flèche 75m - Fixe
Embase 6x6m - Voie de grue
Flèche 40m - Voie de grue
Flèche 45m - Voie de grue
Flèche 50m - Voie de grue
Flèche 55m - Voie de grue
Flèche 60m - Voie de grue
Flèche 65m - Voie de grue
Flèche 70m - Voie de grue
Flèche 75m - Voie de grue

FIGURE 12 EXAMPLE OF CRANE DYNAMIC BLOCK ON AUTOCAD

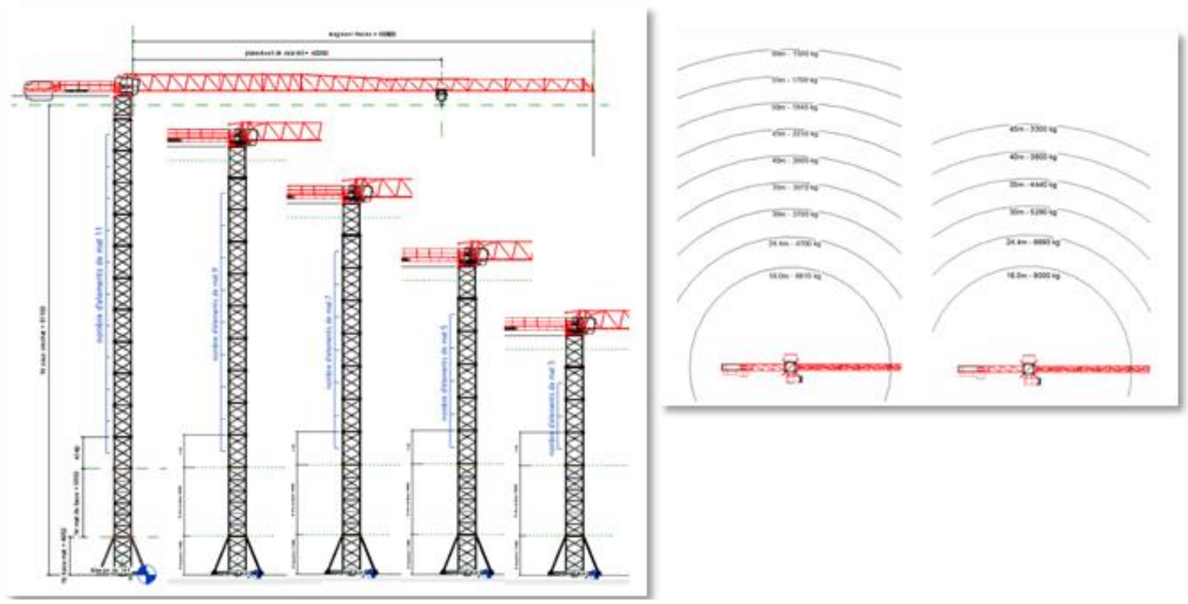


FIGURE 13 ... AND THE 2D EQUIVALENT PARAMETRIC BLOCK ON REVIT

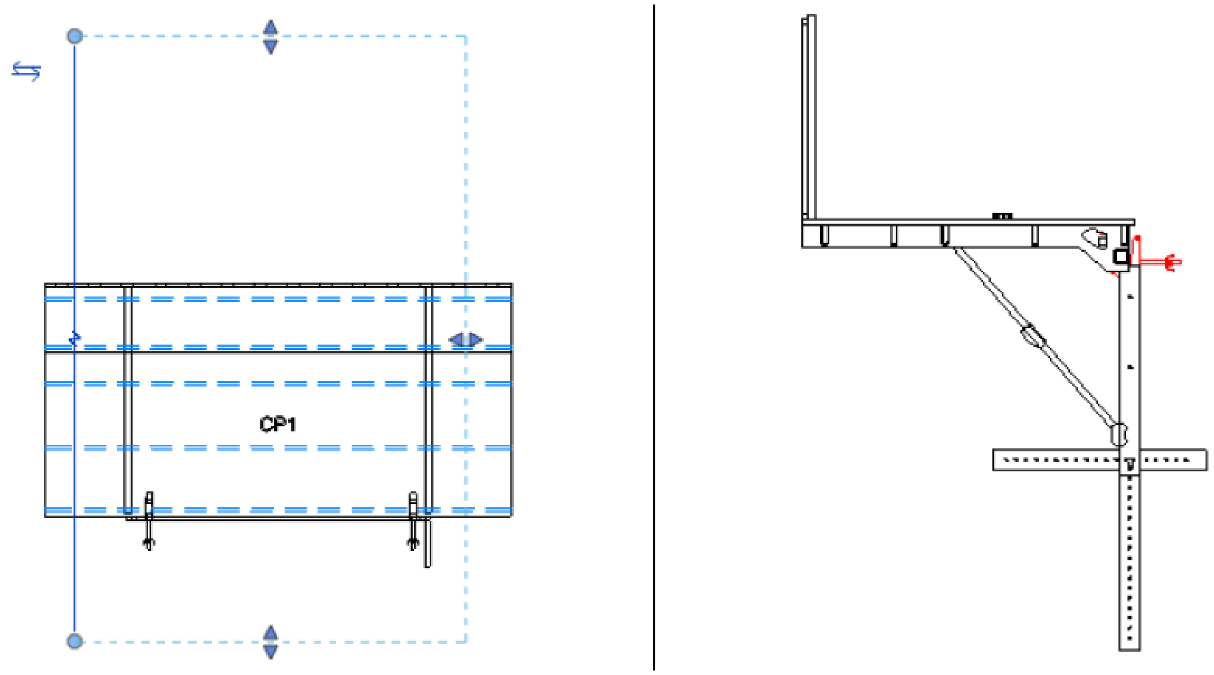


FIGURE 14 SAFETY PLATFORM IN PLAN AND SECTION VIEW

3 The ability to have 3 levels of details

This is very useful when we deal with a lot of components in one single Revit file (a typical example is the day by day formwork sequence). It offers a good way to work with many blocks in the drawing without performance costing in Revit. It is also a good way to switch from a very simple view (with only the details we are interested in) to a more realistic view for on-site communication. The wall formwork family is a typical example: in coarse view, we only want to show the holes made tightening devices so that we are able to see their exact positions.

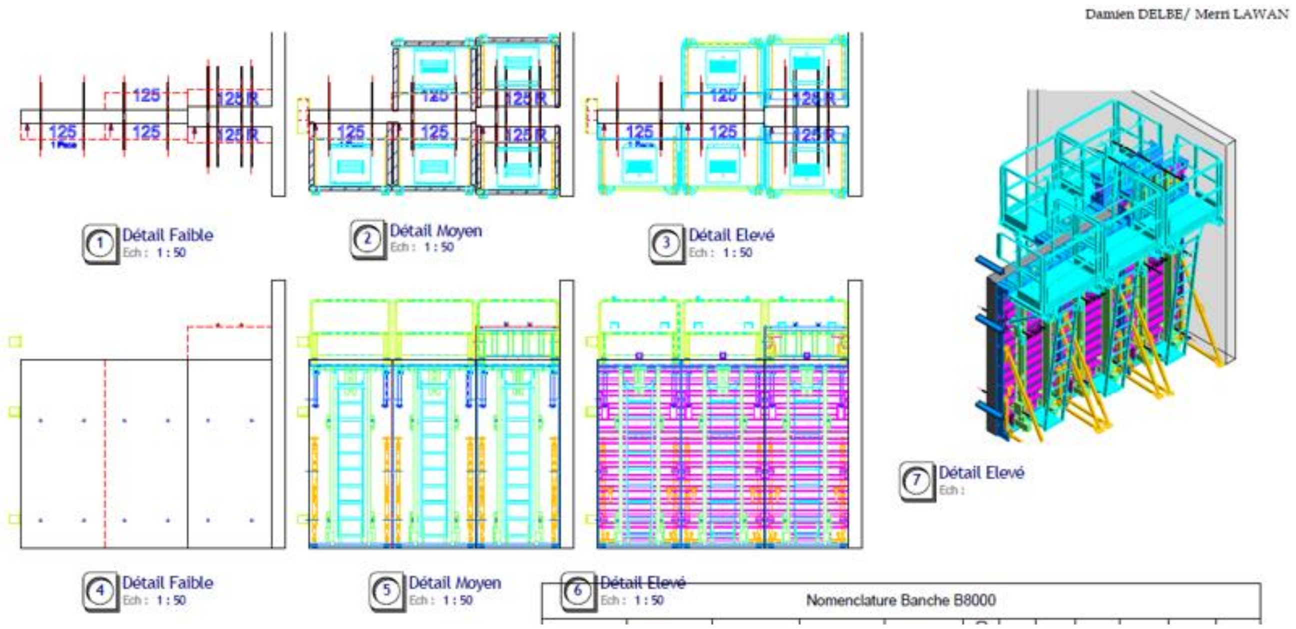


FIGURE 15 DIFFERENT DETAILS LEVELS SHOWING DIFFERENT INFORMATION

4 AutoCAD compatibility in 2D and 3D

When a DWG drawing is imported and exploded (and it is highly recommended to so if you don't want to have performance issues and if you want to be able to assign different materials on your different volumes) into a Revit family file, we can get AutoCAD layers – which become Revit Object styles – and retrieve 2D and sometimes 3D geometry (works only if you have 3D geometry which appears as "Solid objects" in AutoCAD). These features help our Methods department to quickly build a Revit Library from their CAD library.

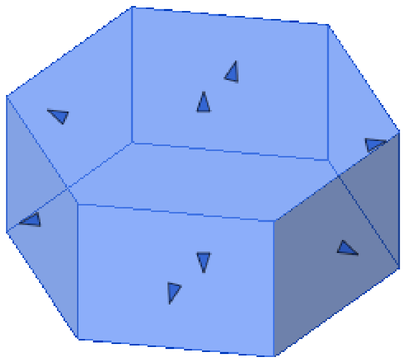
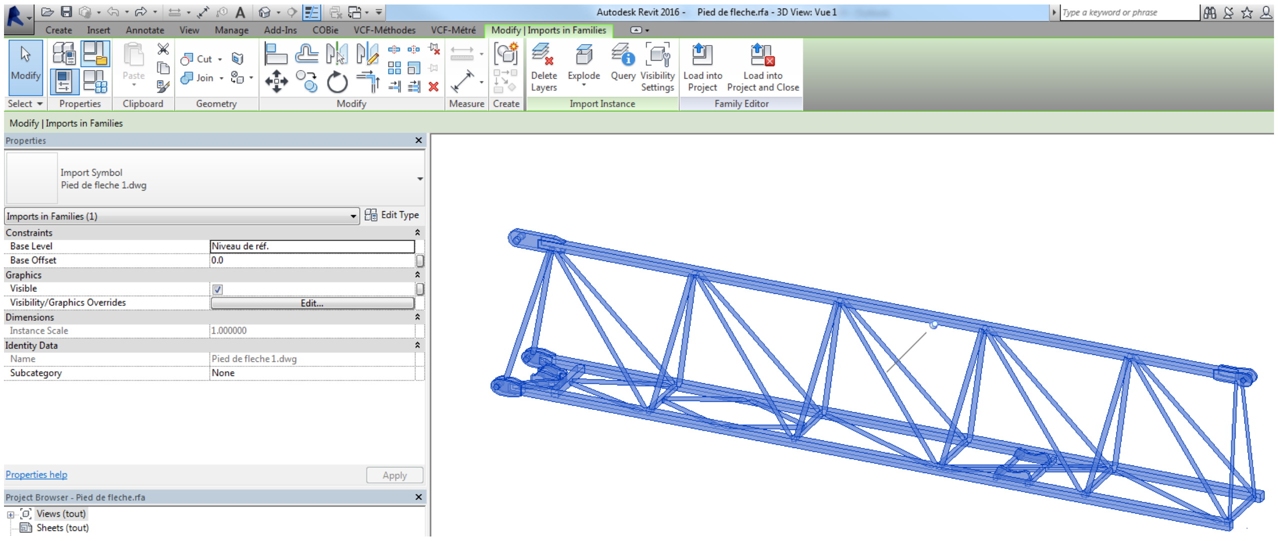


FIGURE 16 SOLID 3D ARE KEPT AS FREE FORM VOLUME WITH HANDLES WHEN EXPLODED IN REVIT


Autodesk Revit 2016 - Pied de fleche.rfa - 3D View: Vue 1
Modify | Imports in Families
Delete Layers
Explode
Query
Visibility Settings
Load into Project
Load into Project and Close
Import Instance
Family Editor

Properties

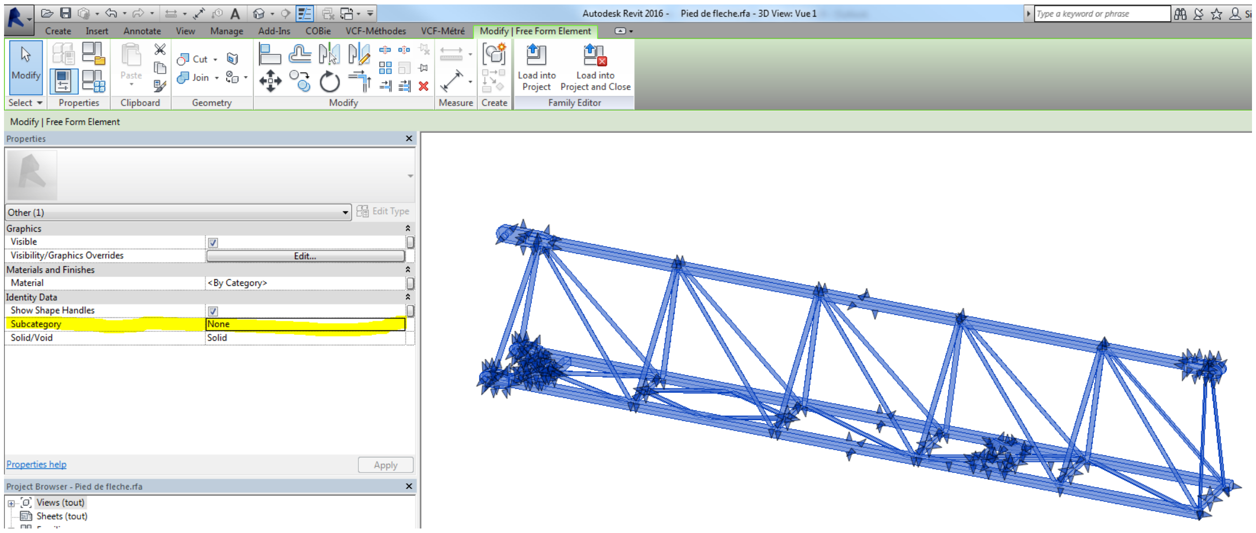
Import Symbol
Pied de fleche 1.dwg

Imports in Families (1)

Constraints	
Base Level	Niveau de réf.
Base Offset	0.0
Graphics	
Visible	✓
Visibility/Graphics Overrides	Edit...
Dimensions	
Instance Scale	1.000000
Identity Data	
Name	Pied de fleche 1.dwg
Subcategory	None

Project Browser - Pied de fleche.rfa
Views (tout)
Sheets (tout)

FIGURE 17 DWG DRAWING BEFORE WE APPLY EXPLODE COMMAND (IT IS ONE SINGLE IMPORTED SYMBOL)


Autodesk Revit 2016 - Pied de fleche.rfa - 3D View: Vue 1
Modify | Free Form Element
Load into Project
Load into Project and Close
Family Editor

Properties

Other (1)

Graphics	
Visible	✓
Visibility/Graphics Overrides	Edit...
Materials and Finishes	
Material	<By Category>
Identity Data	
Show Shape Handles	✓
Subcategory	None
Solid/Void	Solid

Project Browser - Pied de fleche.rfa
Views (tout)
Sheets (tout)

FIGURE 18 SEVERAL 3D WHICH ENABLE US TO APPLY SEPARATE OBJECTS STYLES

5 Family nesting feature enables us to breakdown complex families into simple pieces and make maintenance operations easier

Revit families enable communication between host family and its children, so there is no loss of information. Maintenance is also simplified because if you want to later add more details into a nested family, you can do it without breaking the overall component. On our complex families we often have at least 3 or 4 nested levels

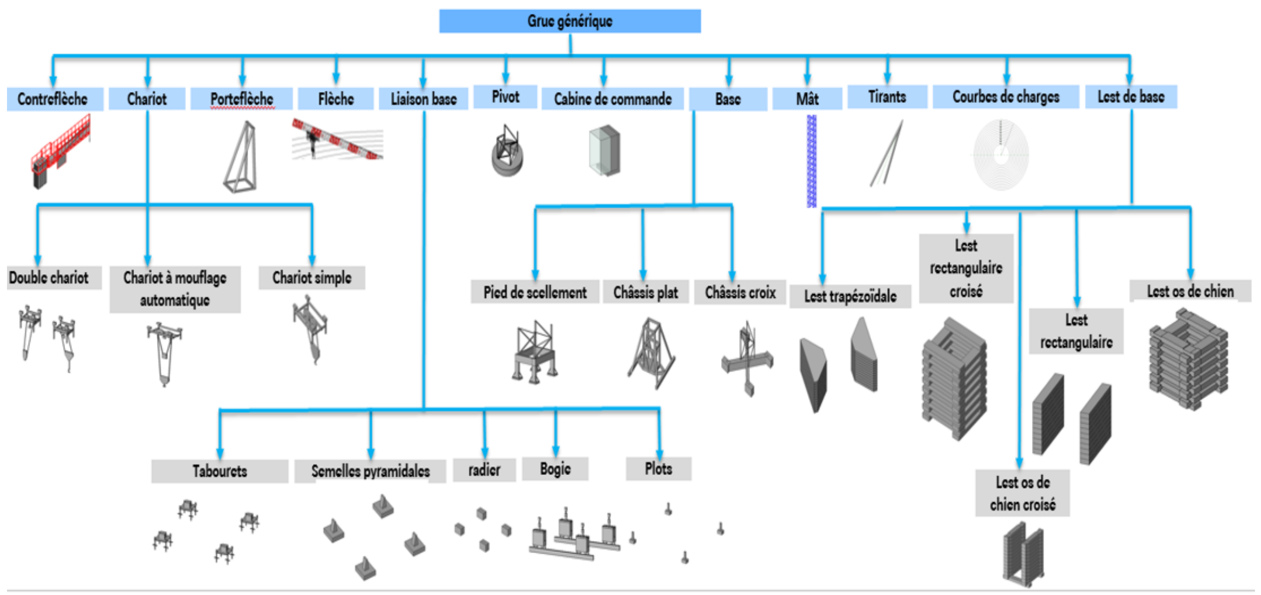
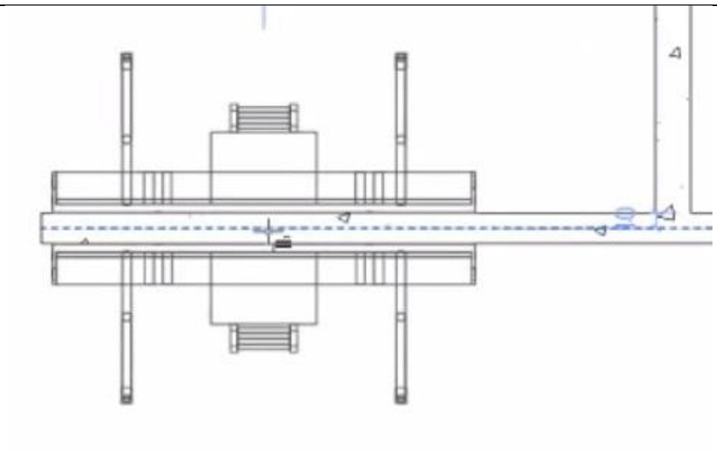
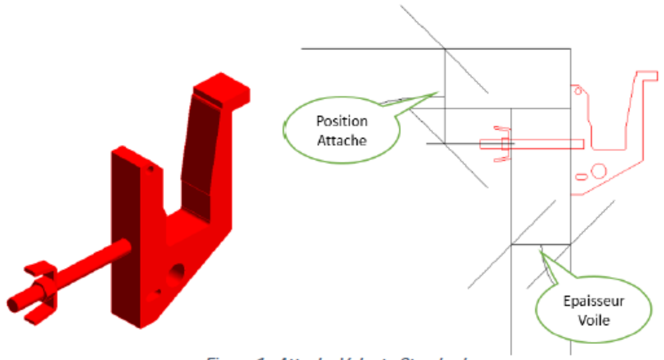
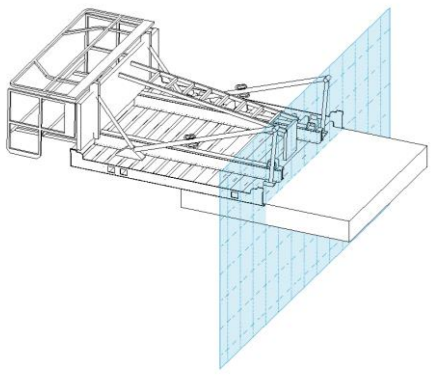


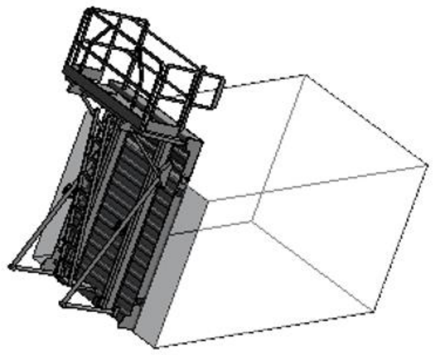
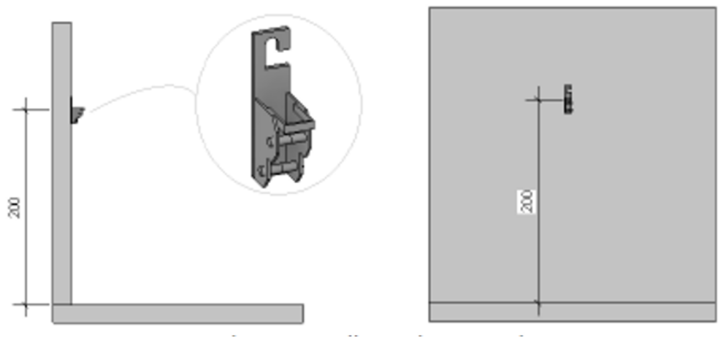
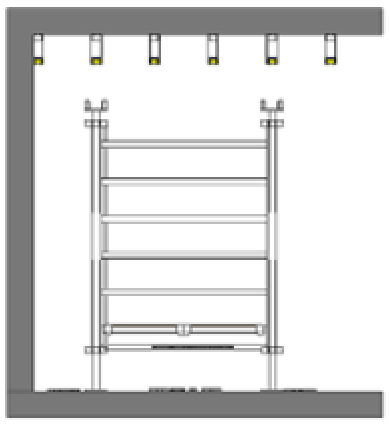
FIGURE 19 IMBRICATION EXAMPLE FOR THE TOWER CRANE FAMILY

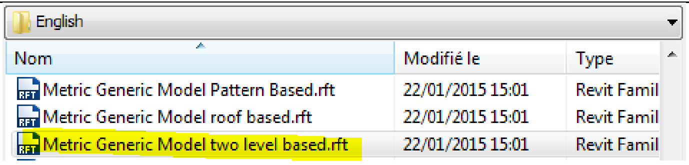
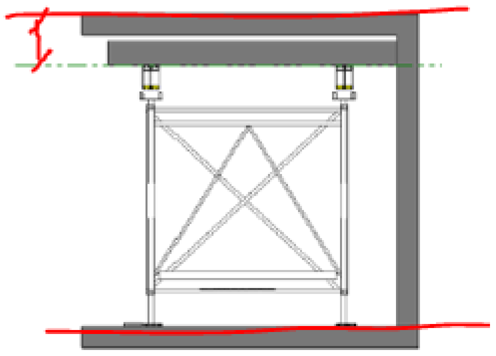
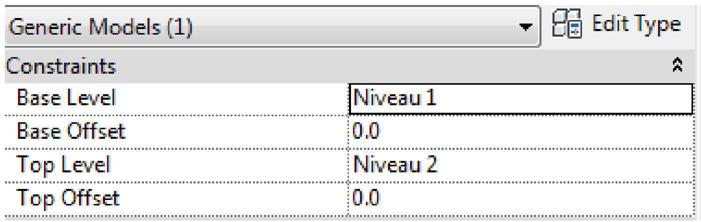
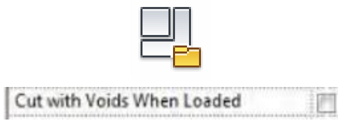
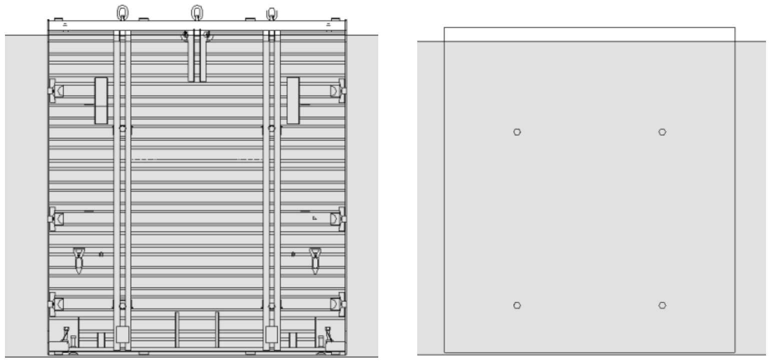
6 Family interactions within the Revit project environment

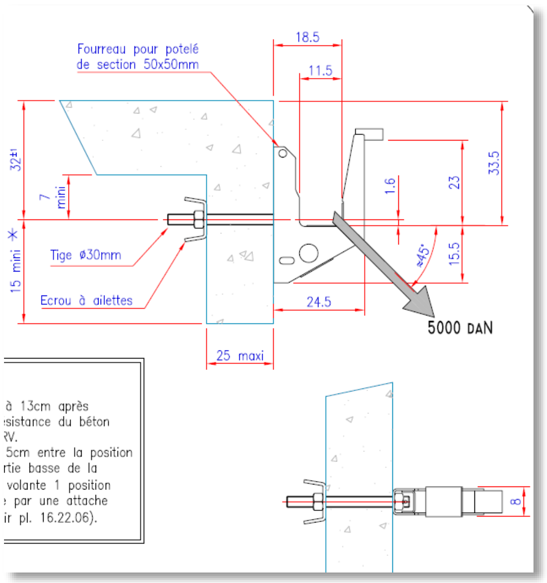
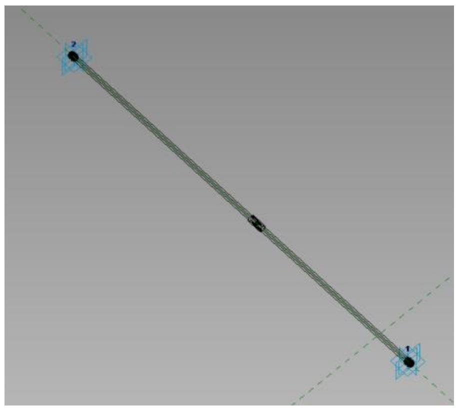
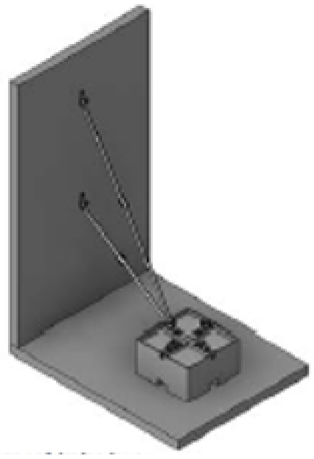
In Revit, there are many interesting features that allow families to interact within the project. Depending on the template we choose we are able to enhance specific materiel component interactions:

Revit Template	Features and usage example
Wall based Metric Generic Model wall based.rft	• Automatically get the thickness from the wall and apply the value to the distance between the two panels within the formwork family. The family is also hosted into the wall which simply the formwork alignment process within the project:

	 • Metal clips insertion on walls is also simplified by this feature:  Figure 1 : Attache Volante Standard
Face based family	• This allow the equipment to align on non-vertical construction plan. This feature can be used to represent inclined formwork: 

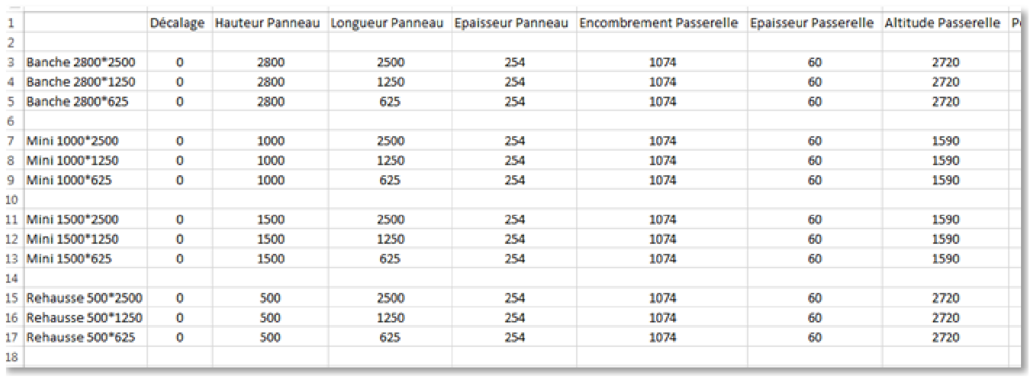
	 • …Or to quickly position push-pull props supports:  • … Or quickly place wood girders under the concrete floor without specify any elevation 
Two level based	The ability to constrain the family on two levels (base and form level): useful for shoring tower layout:

	 English — Nom / Modifié le / Type: Metric Generic Model Pattern Based.rft, 22/01/2015 15:01, Revit Famil; Metric Generic Model roof based.rft, 22/01/2015 15:01, Revit Famil; Metric Generic Model two level based.rft, 22/01/2015 15:01, Revit Famil   Generic Models (1) — Edit Type — Constraints: Base Level: Niveau 1; Base Offset: 0.0; Top Level: Niveau 2; Top Offset: 0.0
Cut with voids when loaded  Cut with Voids When Loaded	• The ability to leave concrete holes due to the tightening devices after placing vertical formwork. When the formwork family is hided, holes are staying so we can precisely make some positions checking  • This can be also useful for structural engineers to get the exact position of the anchoring fasteners and their loads from the Methods' safety layouts drawings:

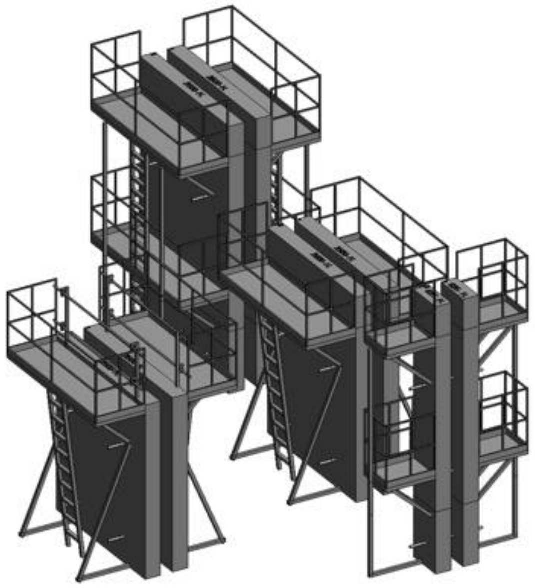
	 Fourreau pour potelé de section 50x50mm 18.5 11.5 32±1 7 mini 15 mini Tige ø30mm Ecrou à ailettes 1.6 23 33.5 ≈45° 15.5 24.5 5000 DAN 25 maxi à 13cm après ésistance du béton RV. 5cm entre la position rtie basse de la volante 1 position par une attache ir pl. 16.22.06). 8
Adaptive components for placing tools	• It allows to place a brace or a push-pull props by two points while respecting the maximum permissible distance for example 

7 Simple database connection:

Type Catalog	This feature enable us to store type parameter values into a txt file. We use it for example to generate different formwork modules from a csv file:

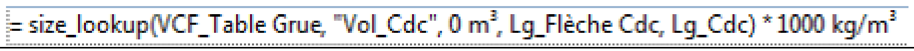


		Décalage	Hauteur Panneau	Longueur Panneau	Epaisseur Panneau	Encombrement Passerelle	Epaisseur Passerelle	Altitude Passerelle	P
1		Décalage	Hauteur Panneau	Longueur Panneau	Epaisseur Panneau	Encombrement Passerelle	Epaisseur Passerelle	Altitude Passerelle	P
2									
3	Banche 2800*2500	0	2800	2500	254	1074	60	2720	
4	Banche 2800*1250	0	2800	1250	254	1074	60	2720	
5	Banche 2800*625	0	2800	625	254	1074	60	2720	
6									
7	Mini 1000*2500	0	1000	2500	254	1074	60	1590	
8	Mini 1000*1250	0	1000	1250	254	1074	60	1590	
9	Mini 1000*625	0	1000	625	254	1074	60	1590	
10									
11	Mini 1500*2500	0	1500	2500	254	1074	60	1590	
12	Mini 1500*1250	0	1500	1250	254	1074	60	1590	
13	Mini 1500*625	0	1500	625	254	1074	60	1590	
14									
15	Rehausse 500*2500	0	500	2500	254	1074	60	2720	
16	Rehausse 500*1250	0	500	1250	254	1074	60	2720	
17	Rehausse 500*625	0	500	625	254	1074	60	2720	
18									

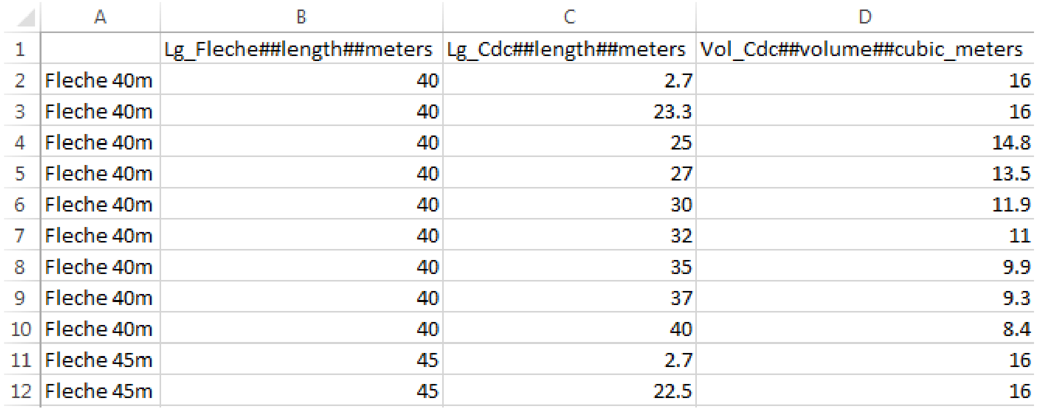


Lookup Tables

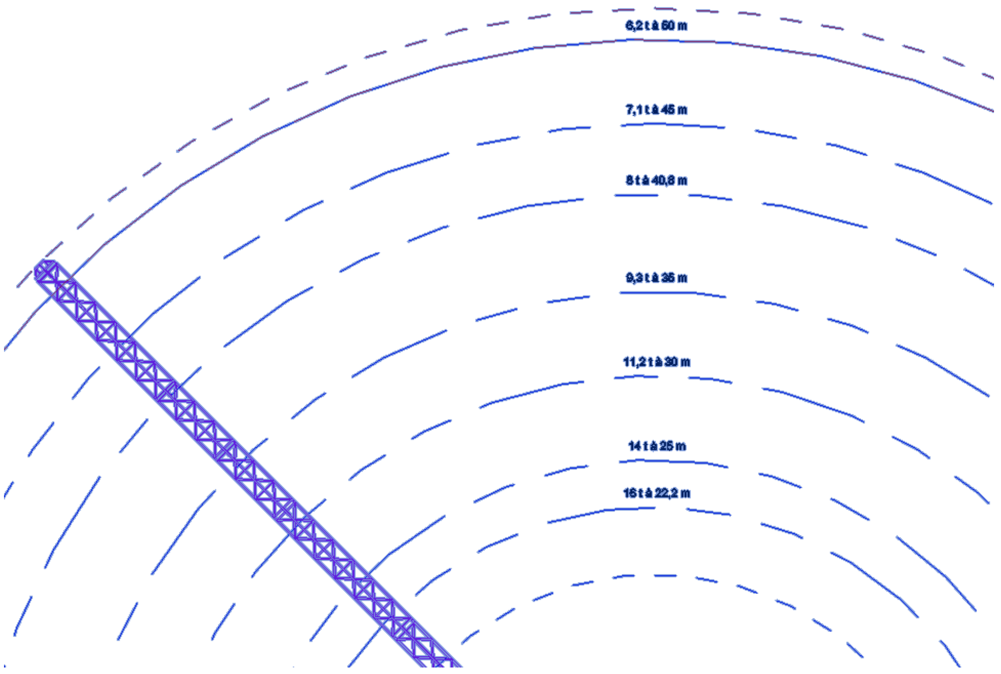
This feature offers the ability to create multiple part sizes without creating a separate family type for each size. Values are defined in a comma separated values csv file. We typically use it to store load curves information according to the jib length and the fork lift position:



= size_lookup(VCF_Table Grue, "Vol_Cdc", 0 m³, Lg_Flèche Cdc, Lg_Cdc) * 1000 kg/m³

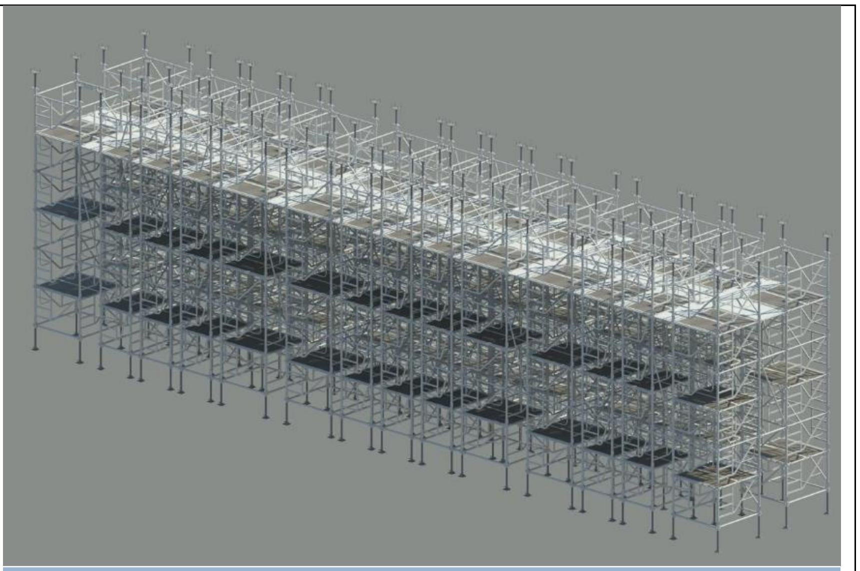


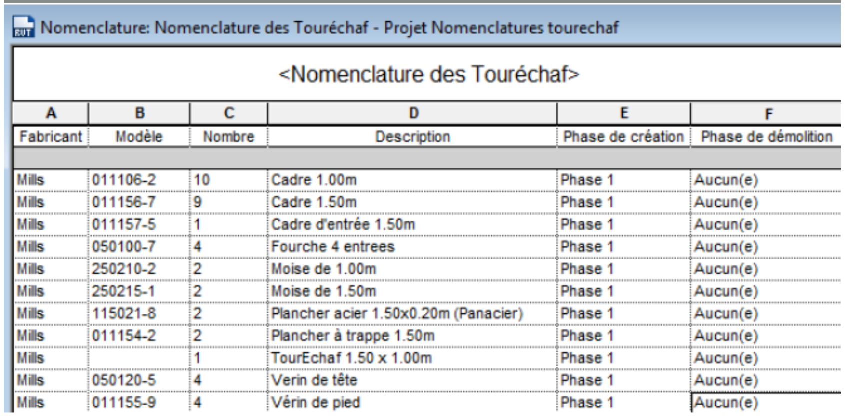
	A	B	C	D
1		Lg_Fleche##length##meters	Lg_Cdc##length##meters	Vol_Cdc##volume##cubic_meters
2	Fleche 40m	40	2.7	16
3	Fleche 40m	40	23.3	16
4	Fleche 40m	40	25	14.8
5	Fleche 40m	40	27	13.5
6	Fleche 40m	40	30	11.9
7	Fleche 40m	40	32	11
8	Fleche 40m	40	35	9.9
9	Fleche 40m	40	37	9.3
10	Fleche 40m	40	40	8.4
11	Fleche 45m	45	2.7	16
12	Fleche 45m	45	22.5	16



8 Scheduling and tagging features

Phasing and schedule capability	• The equipment order is based on the location duration, the user rate and the maximum equipment needed during a phase. By using the Phasing and Schedule feature in Revit, the exact building site needs can calculated:

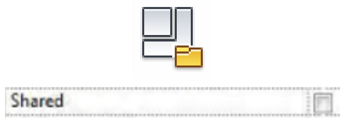
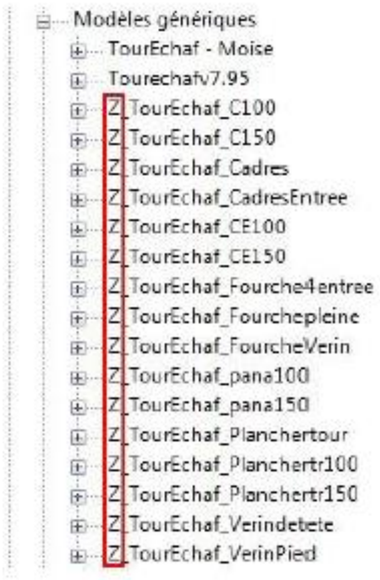


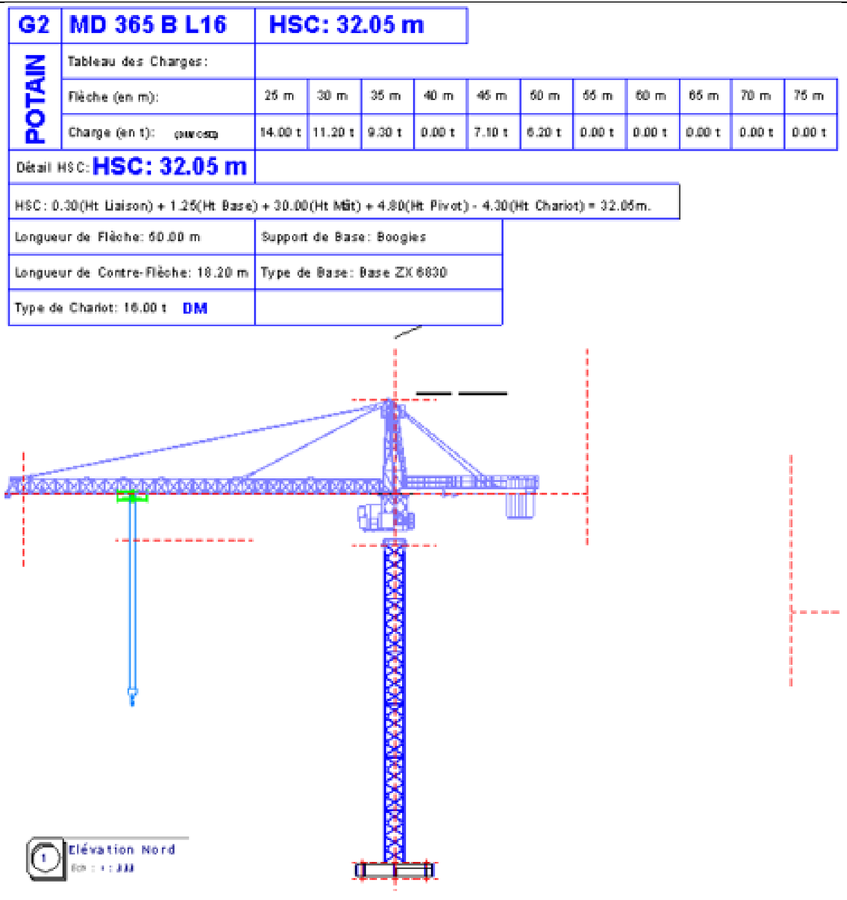


Nomenclature: Nomenclature des Touréchaf - Projet Nomenclatures tourechaf

<Nomenclature des Touréchaf>

A	B	C	D	E	F
Fabricant	Modèle	Nombre	Description	Phase de création	Phase de démolition
Mills	011106-2	10	Cadre 1.00m	Phase 1	Aucun(e)
Mills	011156-7	9	Cadre 1.50m	Phase 1	Aucun(e)
Mills	011157-5	1	Cadre d'entrée 1.50m	Phase 1	Aucun(e)
Mills	050100-7	4	Fourche 4 entrees	Phase 1	Aucun(e)
Mills	250210-2	2	Moise de 1.00m	Phase 1	Aucun(e)
Mills	250215-1	2	Moise de 1.50m	Phase 1	Aucun(e)
Mills	115021-8	2	Plancher acier 1.50x0.20m (Panacier)	Phase 1	Aucun(e)
Mills	011154-2	2	Plancher à trappe 1.50m	Phase 1	Aucun(e)
Mills		1	TourEchaf 1.50 x 1.00m	Phase 1	Aucun(e)
Mills	050120-5	4	Verin de tête	Phase 1	Aucun(e)
Mills	011155-9	4	Vérin de pied	Phase 1	Aucun(e)

Ability to share nested families in order to schedule them  Shared	If you check “Shared” property on nested families, it makes them visible inside the Project so we are able to schedule them  Modèles génériques TourEchaf - Moise Tourechafv7.95 Z TourEchaf_C100 Z TourEchaf_C150 Z TourEchaf_Cadres Z TourEchaf_CadresEntree Z TourEchaf_CE100 Z TourEchaf_CE150 Z TourEchaf_Fourche4entree Z TourEchaf_Fourchepleine Z TourEchaf_FourcheVerin Z TourEchaf_pana100 Z TourEchaf_pana150 Z TourEchaf_Planchertour Z TourEchaf_Planchertr100 Z TourEchaf_Planchertr150 Z TourEchaf_Verindetete Z TourEchaf_VerinPied Tip: it is a good usage to prefix nested elements so that they can be distinguished in the project browser
Use Shared parameter to schedule and tag within the Project	The tag feature is very useful when you want to indicate on execution drawings the crane characteristics for example:

G2	MD 365 B L16	HSC: 32.05 m										
POTAIN	Tableau des Charges:											
	Flèche (en m):	25 m	30 m	35 m	40 m	45 m	50 m	55 m	60 m	65 m	70 m	75 m
	Charge (en t):	14.00 t	11.20 t	9.30 t	0.00 t	7.10 t	6.20 t	0.00 t	0.00 t	0.00 t	0.00 t	0.00 t

Détail HSC: **HSC: 32.05 m**

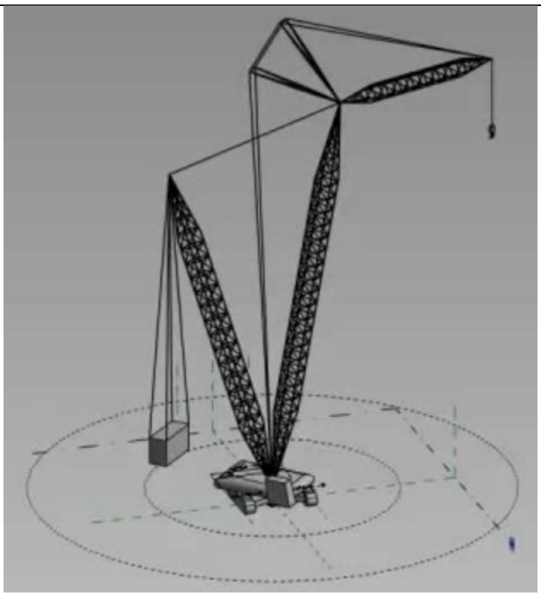
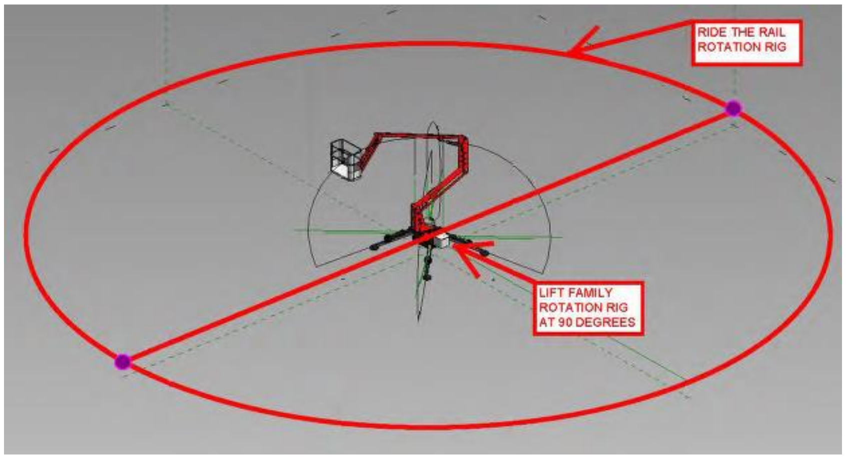
HSC : 0.30(Ht Liaison) + 1.25(Ht Base) + 30.00(Ht Mât) + 4.80(Ht Pivot) - 4.30(Ht Chariot) = 32.05m.

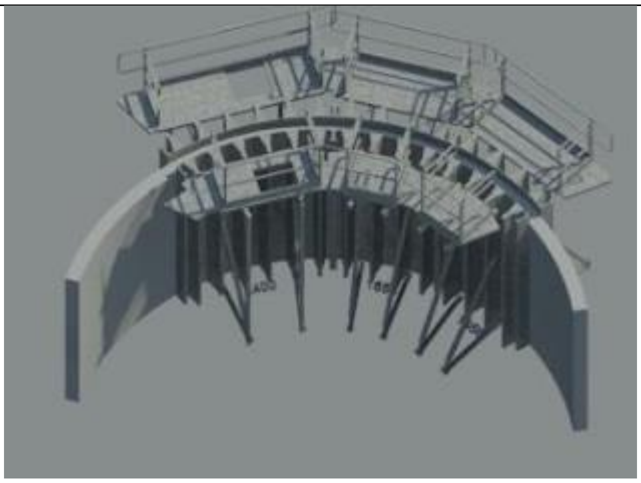
Longueur de Flèche: 60.00 m	Support de Base: Boogies
Longueur de Contre-Flèche: 18.20 m	Type de Base: Base ZX 6030
Type de Chariot: 16.00 t DM	

9 And other modeling and constraints capabilities

I won't go in detail for this topic (I can talk about this many hours), but you can find very interesting tutorials on different blogs or AU or RTC learning section (I quote below some very interesting example I found):

Adaptive components to create articulated arms with specific constraints	• Adaptive big crane example (thanks to Julien BENOIT, you can find a video tutorial on his blog: https://aecuandme.wordpress.com/)

	 • Construction lift family using the "ride the rail method" (thanks to Marcello Sgambelluri, RTC North America 2012)  RIDE THE RAIL ROTATION RIG LIFT FAMILY ROTATION RIG AT 90 DEGREES
And other modeling and constraints capabilities:	Curve formworks, yes it is possible! You can contact me if you are interested in, I may organize a lab next year:

10 And Finally the API connection

Thanks to the Revit API, we can benefit from the .NET Framework in order to:

- Make very friendly user interface (WinForms or WPF)
- Reduce formula complexities inside families
- Easy link your custom families with more powerful database (SQL Server, Access) than a .txt or .csv files.

2.5 The global road map

When the road map for each custom building site tool family has been established, we had to deal with the following common constraints:

- The family has to be realistic in 3D and schematic in 2D
- The geometrical envelope of the family must comply with the exact manufacturer dimensions: it is important when for example a crane is located into building infrastructure:

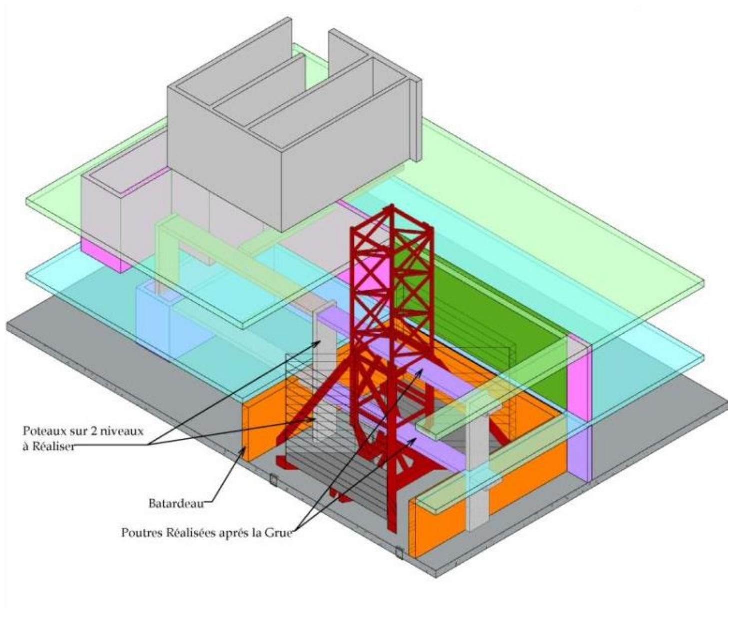


FIGURE 20 CRANE BASE INSTALLATION : EXACT DIMENSIONS ARE CRITICAL

- Full compliance with the company's graphic charter: due to a communication issue, the building site installation – and indirectly the building site tools - must reflect the identity of VINCI CONSTRUCTION FRANCE:

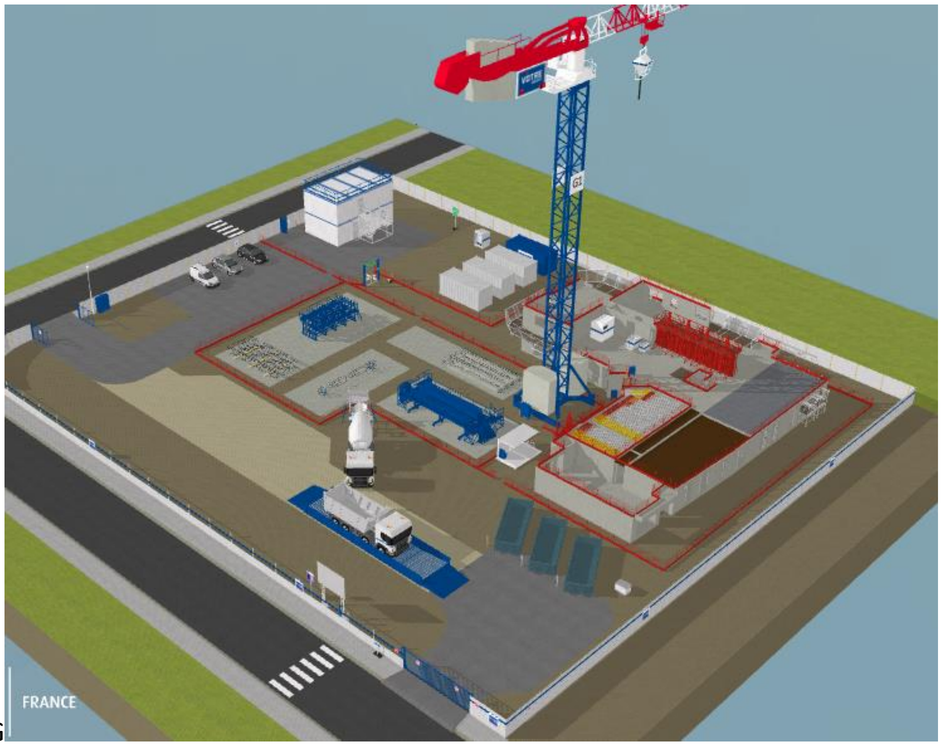
G

FIGURE 21 VINCI'S TYPICAL BUILDING SITE INSTALLATION WITH GRAPHIC COMPLIANCE

- Automatic extraction of all the family parts including their references in order to have a clear vision of what has to be delivered on site

As we were thinking of this, other ideas come into our minds: why not fully use the Revit family's features in order to have more intelligent component and help more our decision making process as well as our drawing process?

As Revit family features offer attributes management to store data (type parameters, lookup tables) we decided to add sets of information like:

- Equipment weight: in order to check if it is liftable by human or crane and check work painfulness
- Setup and takedown time in order to measure the real impact on the planning (great height shoring are particularly time consuming)
- Packing dimensions and space requirement regarding the equipment quantity being delivered: this helps us better define necessary storage area on the building site

As Revit family features also offer the ability of constraining throughout formulas, we decided to integrate as much as possible:

- Manufacturers' rules: equipment dimensions regarding loads for example.

- Internal company's rules: safety rules that are often more constraining than manufacturers' rules

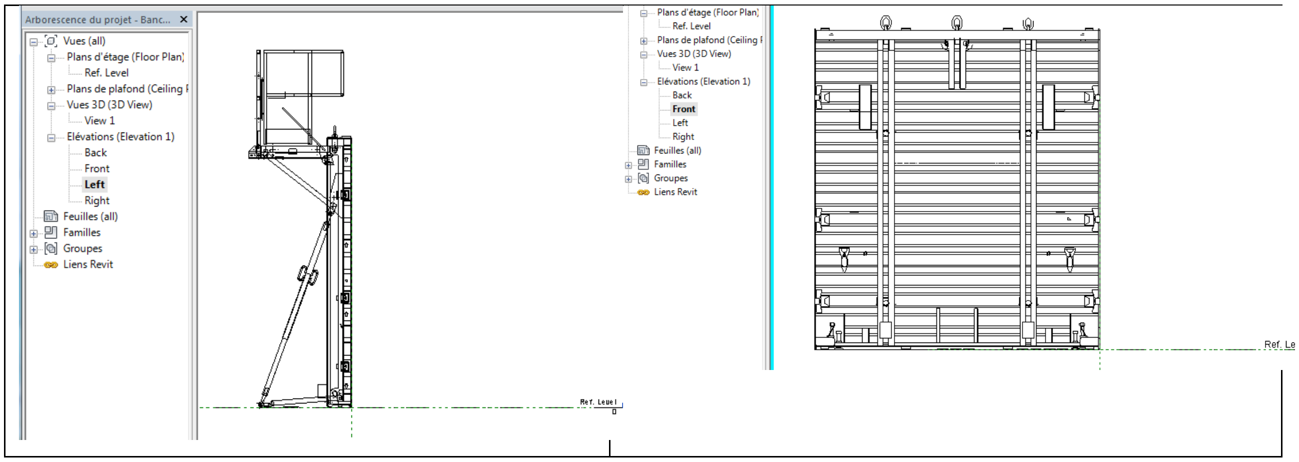
Aware of that, the main risk is going too far in the details. That is why we must find a good balance between the time spent on the library creation and the time saved in productions drawings thanks to the library.

3 Implementing the custom families in Revit

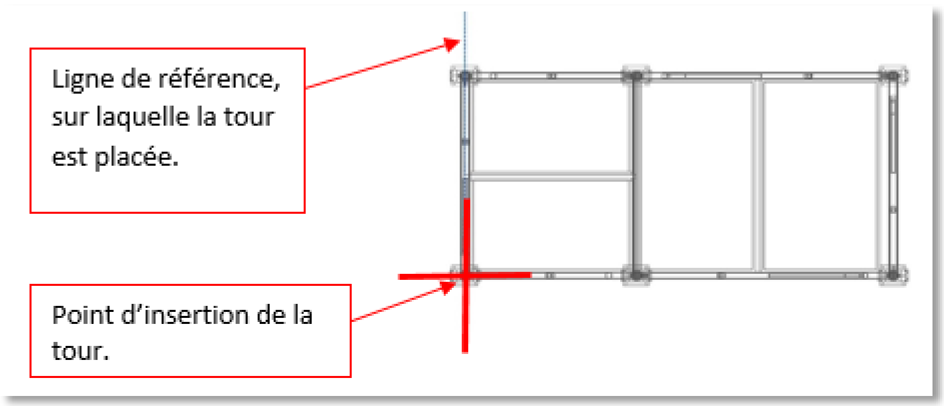
The implanting process leads us to always ask the same questions. These questions must be answered before diving into the family editor. If not the risk is to entirely redo the family if we miss something. In this chapter I will list all the questions we identified as important.

3.1 The detail road map: all the questions to be answered before starting

1. **Family interaction and behavior within the Project:**
   Here is described:
   - **How the insertion point(s) is defined: how many? Positions?** For example, the formwork insertion point will always be on the bottom right in order to make the sheet layout drawing easier :

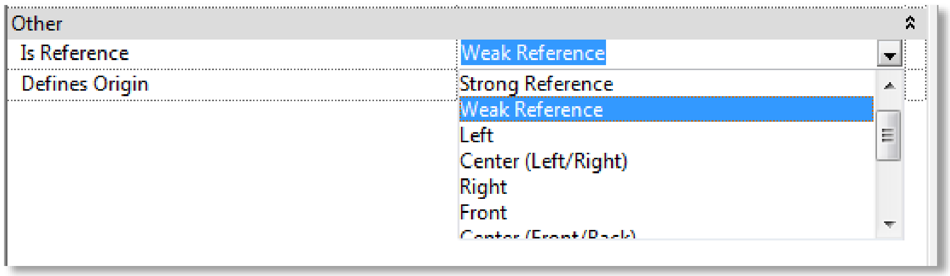


The same logic is also applied for the shoring towers:


Ligne de référence, sur laquelle la tour est placée.
Point d'insertion de la tour.

It is very important not to change over time the insertion point position and keep the logic consistency! Later we plan to develop plugins which will use these reference points in order to helps sheet layout drawings by automatically inserting these families in the project (see final chapter called "What's the next step").

- **Strong or weak reference plane?** In the family editor, which reference plane should be used to annotate or to align on can be specified. This will have a huge impact on your drawing productivity (positioning, annotations):



Other	
Is Reference	Weak Reference
Defines Origin	

FIGURE 22 INDICATION IF THE REFERENCE PLANE WILL BE USED OR NOT WITHIN THE PROJECT

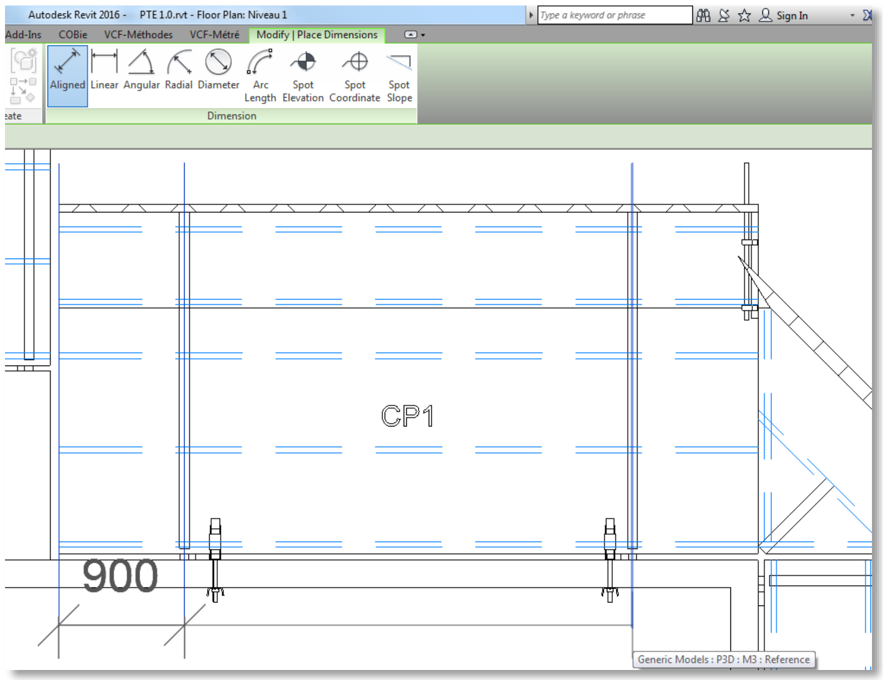
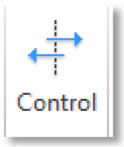
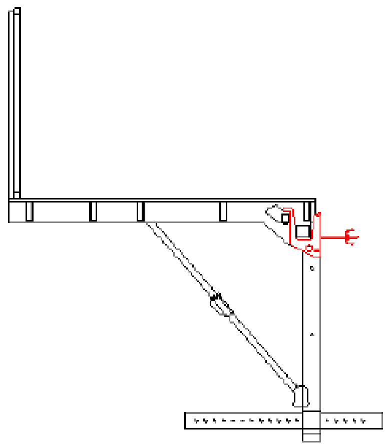
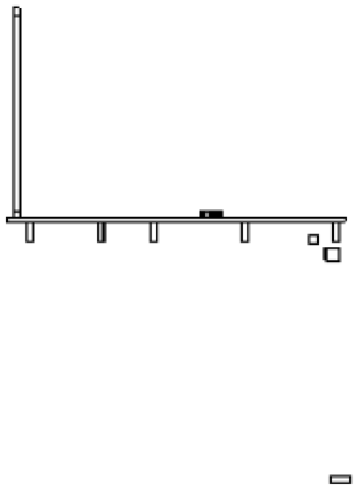


FIGURE 23 ABILITY TO ANNOTATE SAFETY PLATFORM EXTENSION ON THE DESIRED REFERENCE PLANE

- **Add flip controls?** In the family editor, flip controls can also be added which helps user change the orientation of the family on a specified axis within the project

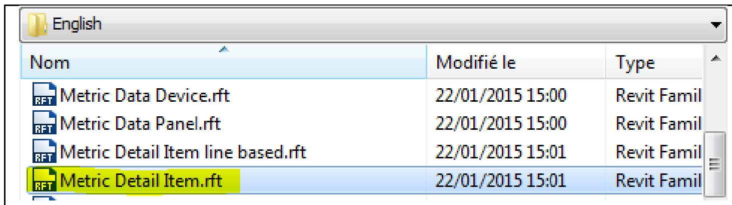
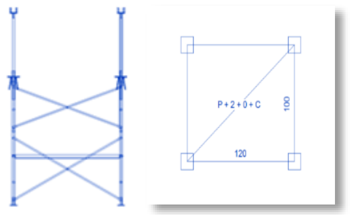


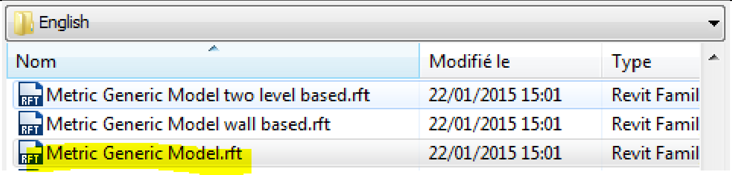
- **Host or face based family?** These features help the family interact with the building components. Host feature is the only way (without the API) to let the family react depending on the host dimensions (wall thickness for example). Because of this interesting feature, we initially chose wall host family for the wall formwork. But we found some limitations: if we want to form a beam, or represent the formwork alone, it is impossible as the family need a wall category host. You also have to be aware that the host feature does not work on linked Project (only face based family does).
- **Which Revit category to choose?** Revit categories are the main internal Revit classification for building components and it helps us in many ways filtering the families within the project (through the view filters, schedules, tags, project browser, calling object commands). Sometimes it also influences how the 3D is cut: for example Mechanical Equipment are never cut while Generic model can be cut.

 FIGURE 24 3D SAFETY PLATFORM NOT CUT IN SECTION PLAN	 FIGURE 25 THE SAME SAFETY PLATFORM CUT

2. Visual level of details in 2D and 3D views

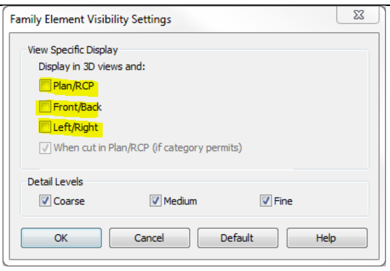
- **Use detail components:** sometimes, it is better to not show the projected 3D geometry in plan and elevation views (because this will lead to too many overlapping geometry). A good practice is to replace 3D geometry by nested detail components – whose special feature is to only be visible in 2D - in 2D views.

 English Nom / Modifié le / Type Metric Data Device.rft / 22/01/2015 15:00 / Revit Famil Metric Data Panel.rft / 22/01/2015 15:00 / Revit Famil Metric Detail Item line based.rft / 22/01/2015 15:01 / Revit Famil Metric Detail Item.rft / 22/01/2015 15:01 / Revit Famil Nesting	This particular template make the geometry only visible in section an plan views (not in 3D view) 
	We check the options that make the 3D geometry only visible in 3D:



English

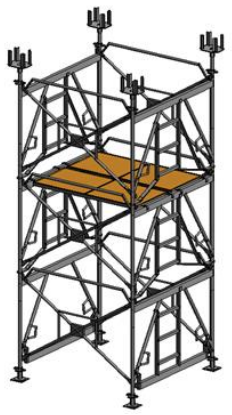
Nom	Modifié le	Type
Metric Generic Model two level based.rft	22/01/2015 15:01	Revit Famil
Metric Generic Model wall based.rft	22/01/2015 15:01	Revit Famil
Metric Generic Model.rft	22/01/2015 15:01	Revit Famil



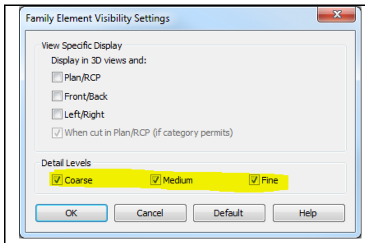
Family Element Visibility Settings

View Specific Display
Display in 3D views and:
- [ ] Plan/RCP
- [ ] Front/Back
- [ ] Left/Right
- [✓] When cut in Plan/RCP (if category permits)

Detail Levels
- [✓] Coarse
- [✓] Medium
- [✓] Fine



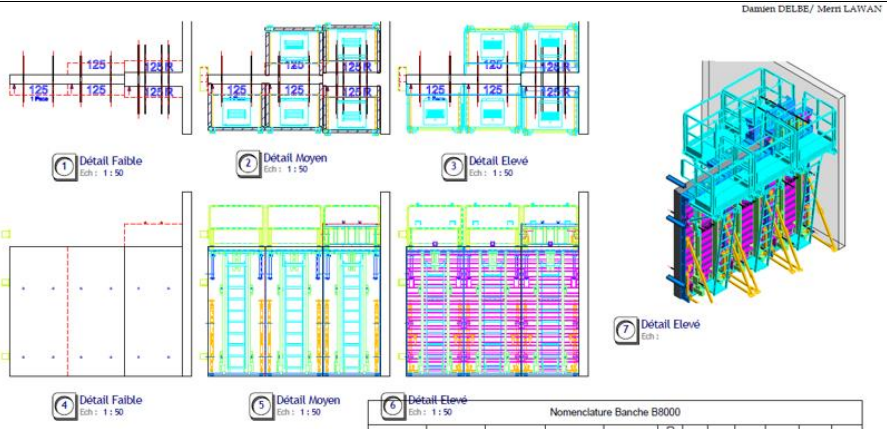
- **Define level of details:** we also define the visual detail level for the 3 Revit scenarios (coarse, medium, fine). This will help create views with only specific details (like wall formwork tightening device positions), and create other views with all the 3D details for communication needs



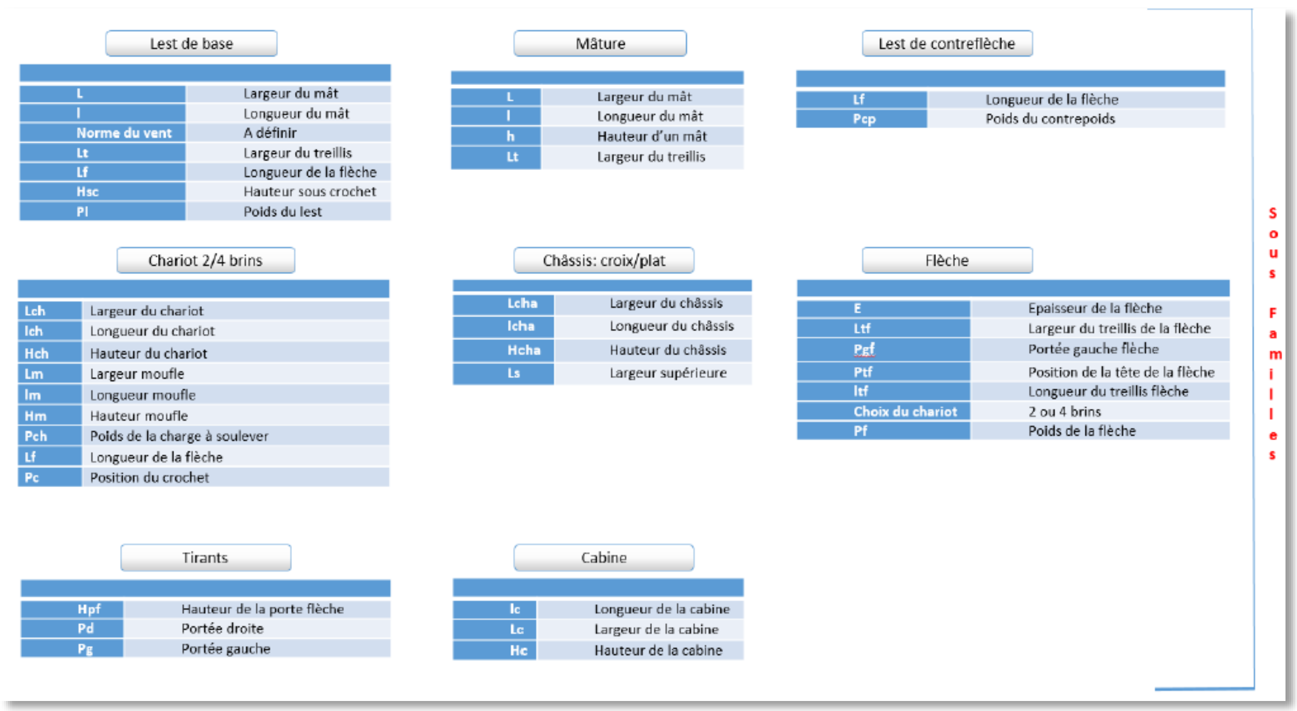
Family Element Visibility Settings

View Specific Display
Display in 3D views and:
- [ ] Plan/RCP
- [ ] Front/Back
- [ ] Left/Right
- [✓] When cut in Plan/RCP (if category permits)

Detail Levels
- [✓] Coarse
- [✓] Medium
- [✓] Fine



3. **Geometry constraining parameters definition**
   All parameters which have a direct influence on geometry constraining and visibility. At this step, it is important to carefully – take time to weigh the pros and cons - choose between instance and type parameter (this will have a big influence on the project side and on the family structure). The naming convention is also very important (tips: do not use space or characters that can be used in formulas).


Sous Familles

Lest de base

L	Largeur du mât
l	Longueur du mât
Norme du vent	A définir
Lt	Largeur du treillis
Lf	Longueur de la flèche
Hsc	Hauteur sous crochet
Pl	Poids du lest

Mâture

L	Largeur du mât
l	Longueur du mât
h	Hauteur d'un mât
Lt	Largeur du treillis

Lest de contreflèche

Lf	Longueur de la flèche
Pcp	Poids du contrepoids

Chariot 2/4 brins

Lch	Largeur du chariot
lch	Longueur du chariot
Hch	Hauteur du chariot
Lm	Largeur moufle
lm	Longueur moufle
Hm	Hauteur moufle
Pch	Poids de la charge à soulever
Lf	Longueur de la flèche
Pc	Position du crochet

Châssis: croix/plat

Lcha	Largeur du châssis
lcha	Longueur du châssis
Hcha	Hauteur du châssis
Ls	Largeur supérieure

Flèche

E	Epaisseur de la flèche
Ltf	Largeur du treillis de la flèche
Pgf	Portée gauche flèche
Ptf	Position de la tête de la flèche
ltf	Longueur du treillis flèche
Choix du chariot	2 ou 4 brins
Pf	Poids de la flèche

Tirants

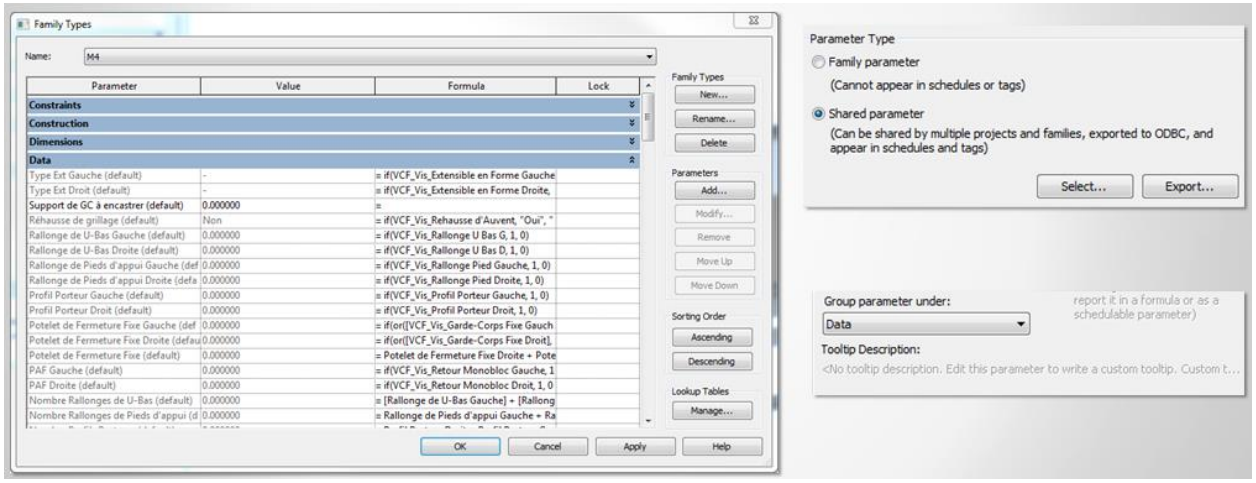
Hpf	Hauteur de la porte flèche
Pd	Portée droite
Pg	Portée gauche

Cabine

lc	Longueur de la cabine
Lc	Largeur de la cabine
Hc	Hauteur de la cabine

FIGURE 26 NESTED COMPONENTS AND THEIR SIZING PARAMETERS

4. **Data parameter definition**
   All parameters not related to geometry but being used to get schedulable and tags information. At this step, as I said previously, it is important to choose between instance and type parameter and define with consistency a naming convention.



Family Types

Name: M4

Parameter	Value	Formula	Lock
Constraints			
Construction			
Dimensions			
Data			
Type Ext Gauche (default)	-	= if(VCF_Vis_Extensible en Forme Gauche	
Type Ext Droit (default)	-	= if(VCF_Vis_Extensible en Forme Droite,	
Support de GC à encastrer (default)	0.000000	=	
Réhausse de grillage (default)	Non	= if(VCF_Vis_Rehausse d'Auvent, "Oui", "	
Rallonge de U-Bas Gauche (default)	0.000000	= if(VCF_Vis_Rallonge U Bas G, 1, 0)	
Rallonge de U-Bas Droite (default)	0.000000	= if(VCF_Vis_Rallonge U Bas D, 1, 0)	
Rallonge de Pieds d'appui Gauche (def	0.000000	= if(VCF_Vis_Rallonge Pied Gauche, 1, 0)	
Rallonge de Pieds d'appui Droite (defa	0.000000	= if(VCF_Vis_Rallonge Pied Droite, 1, 0)	
Profil Porteur Gauche (default)	0.000000	= if(VCF_Vis_Profil Porteur Gauche, 1, 0)	
Profil Porteur Droit (default)	0.000000	= if(VCF_Vis_Profil Porteur Droit, 1, 0)	
Potelet de Fermeture Fixe Gauche (def	0.000000	= if(or([VCF_Vis_Garde-Corps Fixe Gauch	
Potelet de Fermeture Fixe Droite (defau	0.000000	= if(or([VCF_Vis_Garde-Corps Fixe Droit],	
Potelet de Fermeture Fixe (default)	0.000000	= Potelet de Fermeture Fixe Droite + Pote	
PAF Gauche (default)	0.000000	= if(VCF_Vis_Retour Monobloc Gauche, 1	
PAF Droite (default)	0.000000	= if(VCF_Vis_Retour Monobloc Droit, 1, 0	
Nombre Rallonges de U-Bas (default)	0.000000	= [Rallonge de U-Bas Gauche] + [Rallong	
Nombre Rallonges de Pieds d'appui (d	0.000000	= Rallonge de Pieds d'appui Gauche + Ra	

Parameter Type

Family parameter (Cannot appear in schedules or tags)

[✓ Shared parameter] (Can be shared by multiple projects and families, exported to ODBC, and appear in schedules and tags)

report it in a formula or as a schedulable parameter)

Group parameter under: Data

Tooltip Description: <No tooltip description. Edit this parameter to write a custom tooltip. Custom t...

FIGURE 27 DATA SHARED PARAMETER GROUPED UNDER DATA TAB

5. **Schemas which resume the relationship between parameters**
   This will help us to build the parameter formulas in the family editor or to define the general algorithm and user interface logic if we choose to use the API (see API section).

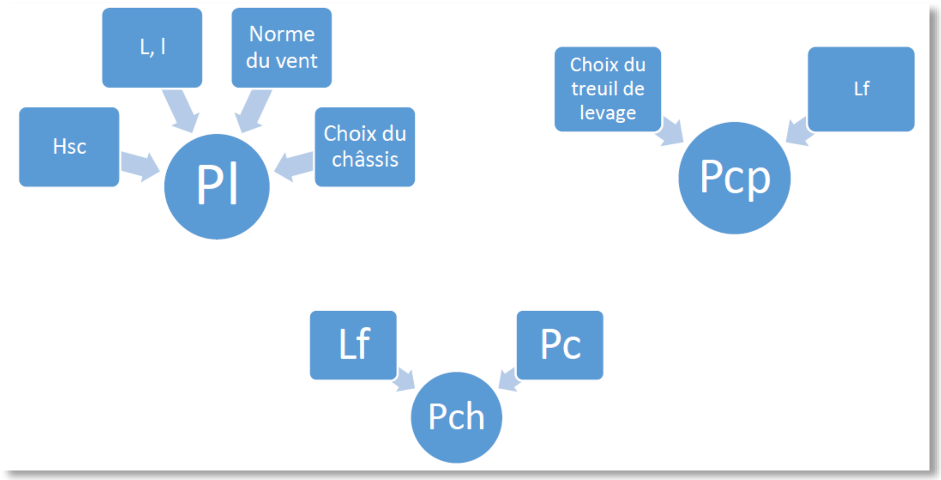


FIGURE 28 RELATIONSHIP BETWEEN PARAMETERS

6. **Separate direct user input parameters from the others**
   This is a very important step. If you do not use an API user interface, you will want to group all these input parameters on the top list and on a separate section in the property panel, so that the user directly clarify his choices. Since the 2015 Revit version, you can also define in read-only all the parameters that are no direct user input.

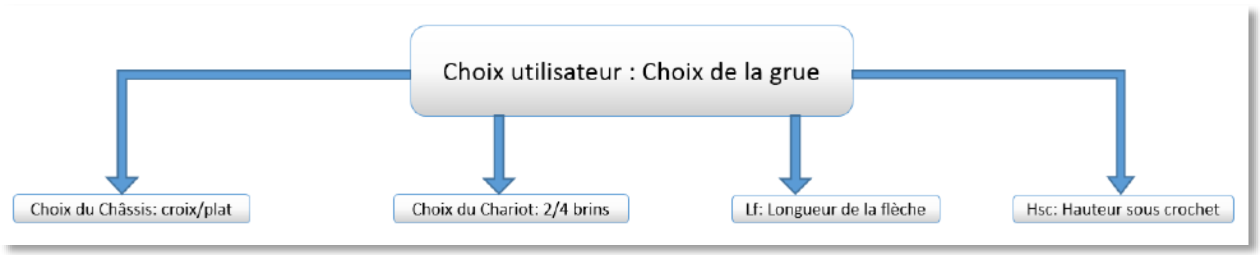

Choix utilisateur : Choix de la grue
Choix du Châssis: croix/plat
Choix du Chariot: 2/4 brins
Lf: Longueur de la flèche
Hsc: Hauteur sous crochet

FIGURE 29 USER DEFINED PARAMETER DIAGRAM

7. **Inventory of all the equipment accessories and manufacturer codification logic integration**
   This step will help us build all the nested components and prepare the overall nested schema.

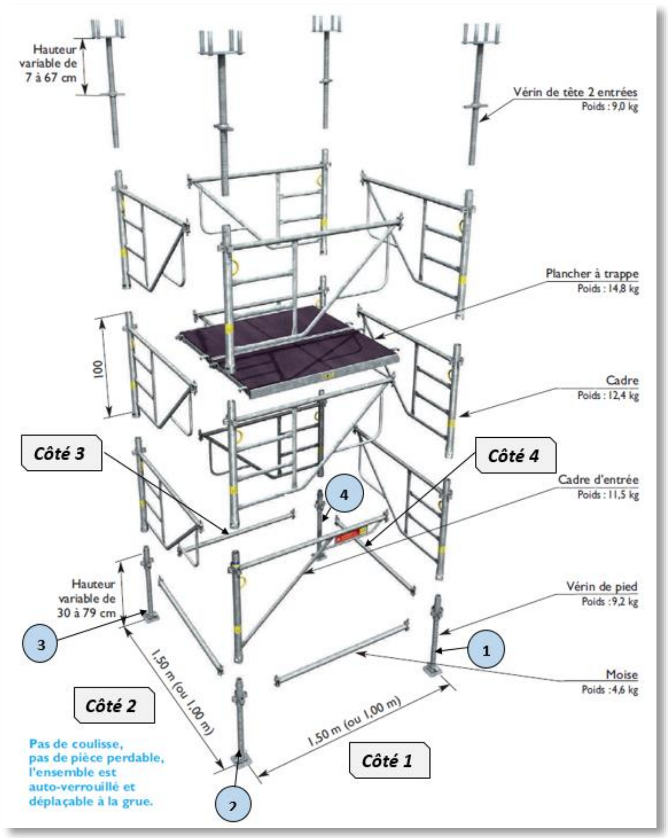
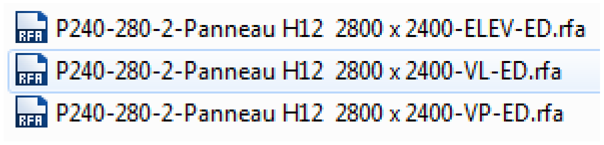


FIGURE 30 ALL THE ACCESSORIES OF A TYPICAL SHORING TOWER

At this stage the naming convention is important as well as a good windows folder classification. It is important to at least write in the file name:

- The module dimensions and reference
- The family template category you use (Detail Item or Generic model for example)
- The view type the component will be used in (3D, elevation or plan view)

- Tips: do not use () but [] instead because otherwise you will get an error when loading the family



P240-280-2-Panneau H12 2800 x 2400-ELEV-ED.rfa
P240-280-2-Panneau H12 2800 x 2400-VL-ED.rfa
P240-280-2-Panneau H12 2800 x 2400-VP-ED.rfa

FIGURE 31 FILE NAMING CONVENTION

It is also important to specify for each nested component an insertion point that should not move overtime (it will help maintenance and further replacement):

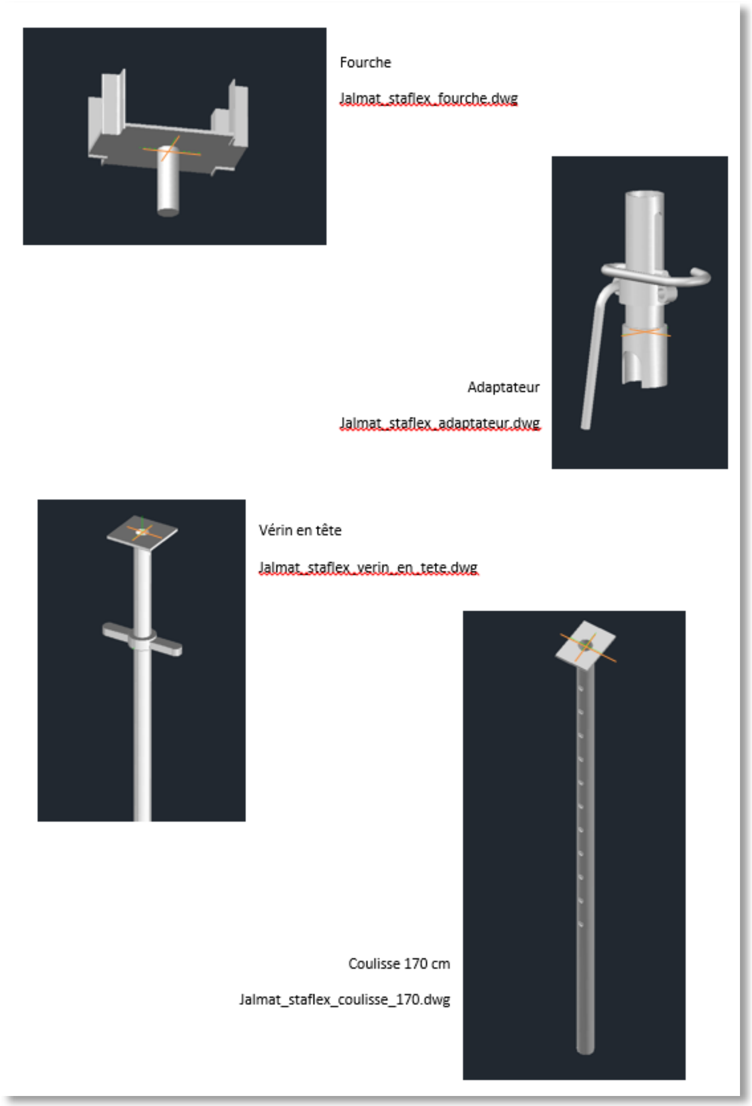
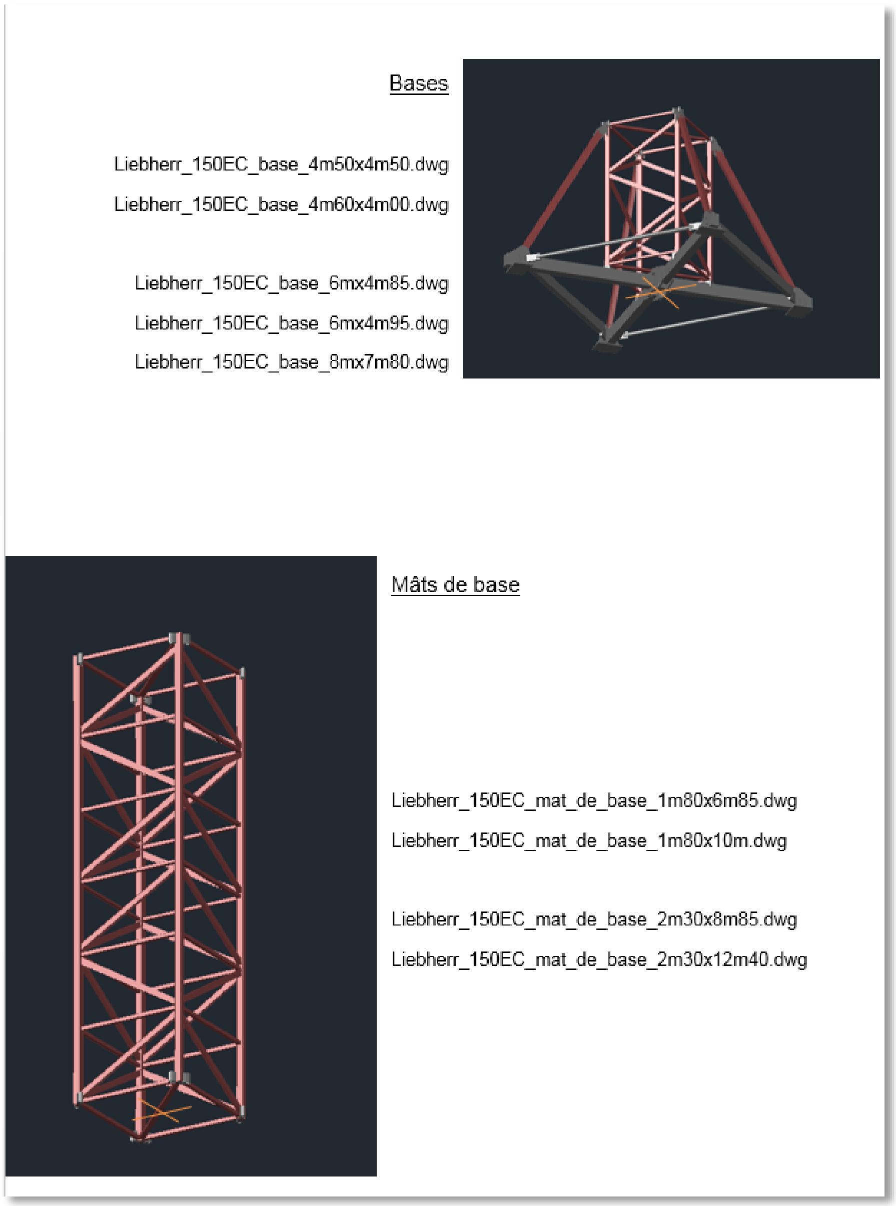

Fourche
Jalmat_staflex_fourche.dwg
Adaptateur
Jalmat_staflex_adaptateur.dwg
Vérin en tête
Jalmat_staflex_verin_en_tete.dwg
Coulisse 170 cm
Jalmat_staflex_coulisse_170.dwg

FIGURE 32 SHORING TOWER ACCESSORIES AND THEIR INSERTION POINT DEFINITION


Bases

Liebherr_150EC_base_4m50x4m50.dwg
Liebherr_150EC_base_4m60x4m00.dwg

Liebherr_150EC_base_6mx4m85.dwg
Liebherr_150EC_base_6mx4m95.dwg
Liebherr_150EC_base_8mx7m80.dwg

Mâts de base

Liebherr_150EC_mat_de_base_1m80x6m85.dwg
Liebherr_150EC_mat_de_base_1m80x10m.dwg

Liebherr_150EC_mat_de_base_2m30x8m85.dwg
Liebherr_150EC_mat_de_base_2m30x12m40.dwg

FIGURE 33 TOWER CRANE ACCESSORIES AND THEIR INSERTION POINT DEFINITION

8. Global nesting schema and parameters linking

We define the global nesting structure and how parameters are linked between host and nested components

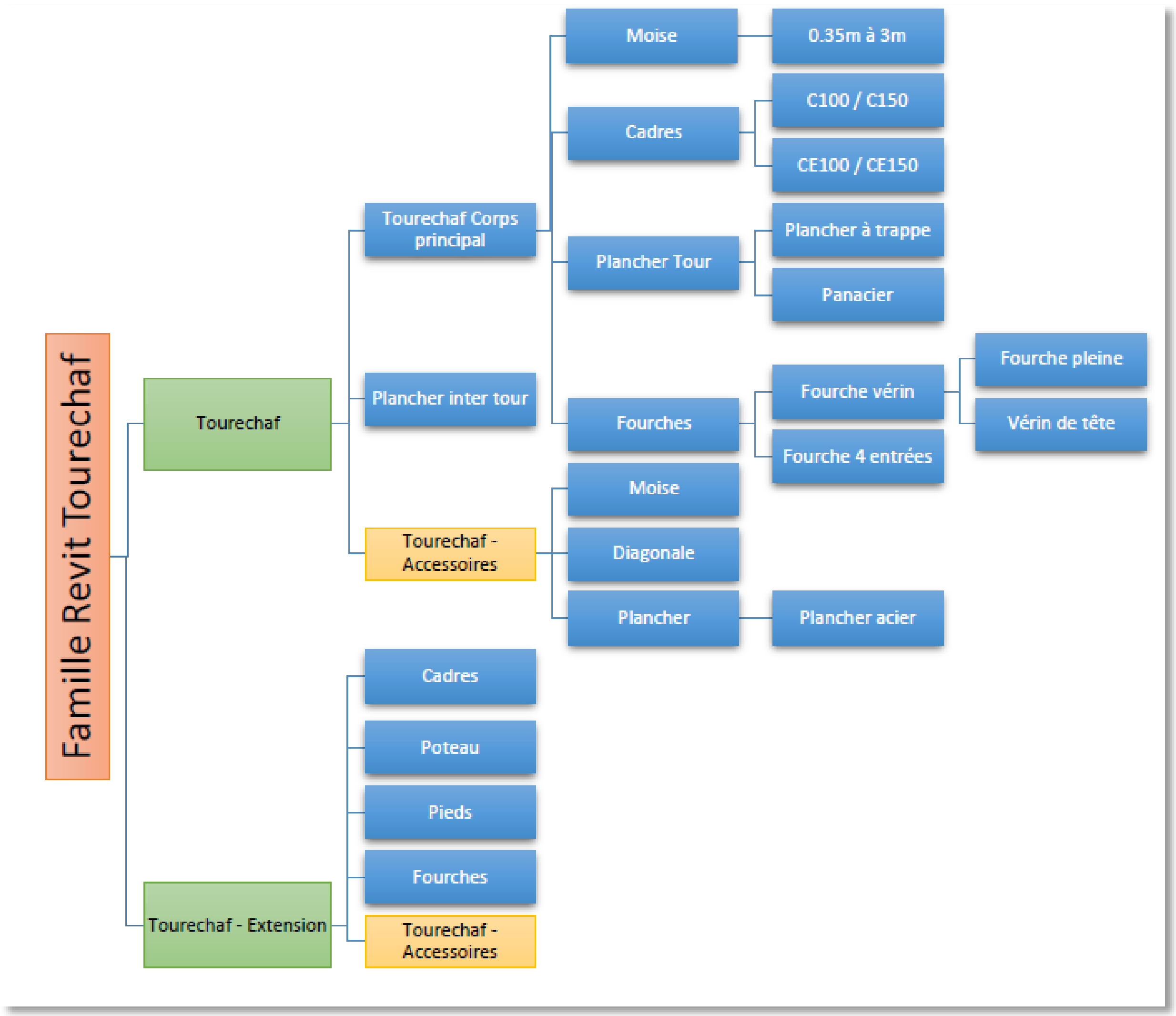
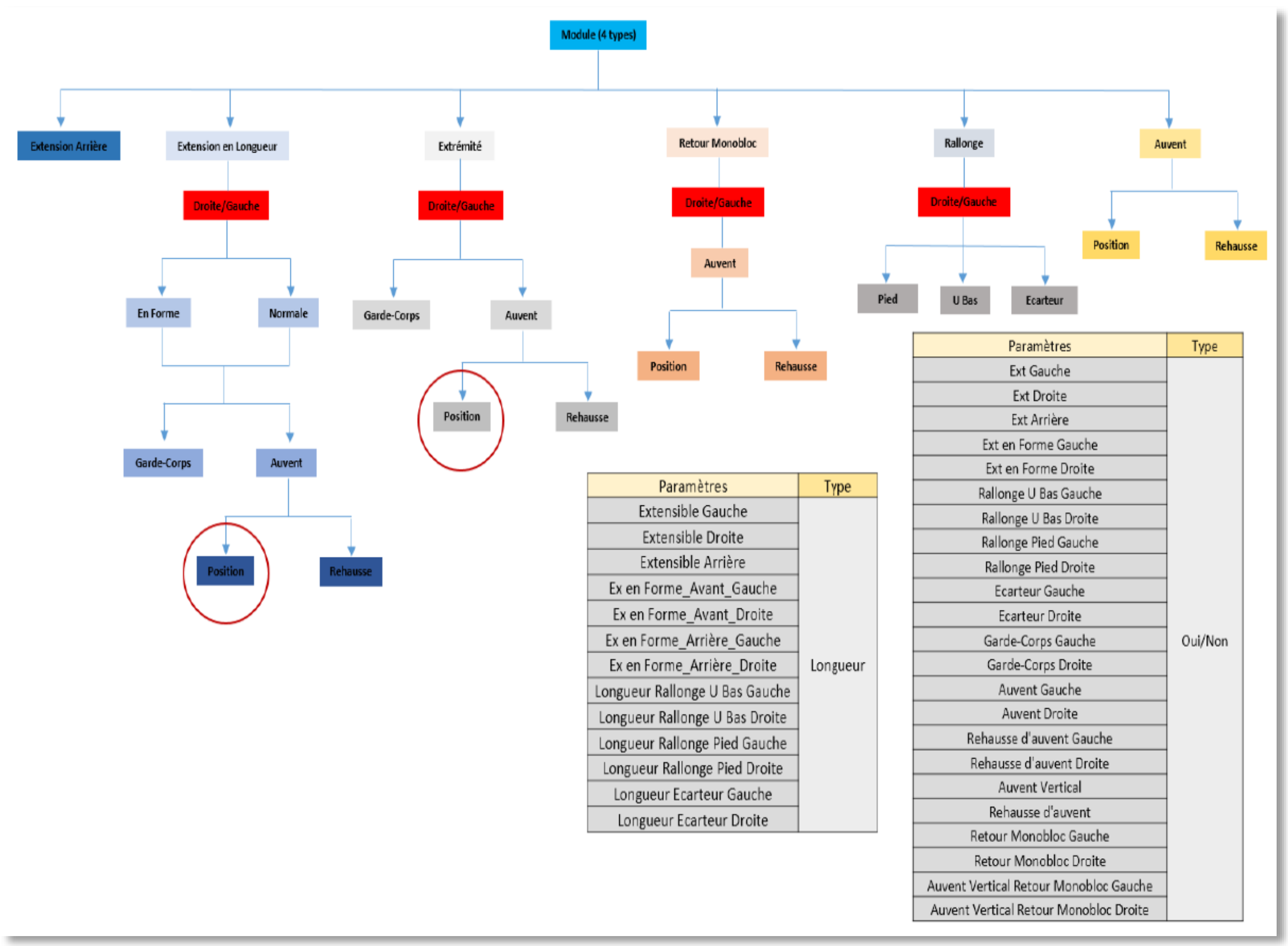


FIGURE 34 GLOBAL NESTING STRUCTURE: SHORING TOWER EXAMPLE



Paramètres	Type
Extensible Gauche	
Extensible Droite	
Extensible Arrière	
Ex en Forme_Avant_Gauche	
Ex en Forme_Avant_Droite	
Ex en Forme_Arrière_Gauche	
Ex en Forme_Arrière_Droite	Longueur
Longueur Rallonge U Bas Gauche	
Longueur Rallonge U Bas Droite	
Longueur Rallonge Pied Gauche	
Longueur Rallonge Pied Droite	
Longueur Ecarteur Gauche	
Longueur Ecarteur Droite	

Paramètres	Type
Ext Gauche	
Ext Droite	
Ext Arrière	
Ext en Forme Gauche	
Ext en Forme Droite	
Rallonge U Bas Gauche	
Rallonge U Bas Droite	
Rallonge Pied Gauche	
Rallonge Pied Droite	
Ecarteur Gauche	
Ecarteur Droite	
Garde-Corps Gauche	Oui/Non
Garde-Corps Droite	
Auvent Gauche	
Auvent Droite	
Rehausse d'auvent Gauche	
Rehausse d'auvent Droite	
Auvent Vertical	
Rehausse d'auvent	
Retour Monobloc Gauche	
Retour Monobloc Droite	
Auvent Vertical Retour Monobloc Gauche	
Auvent Vertical Retour Monobloc Droite	

FIGURE 35 GLOBAL NESTING STRUCTURE: SAFETY PLATFORM EXAMPLE

3.2 Two different ways to build a custom family

During our implementation process into Revit we identified two opposite ways to build a family, both with the pros and the cons:

- The first way is what I like to call **"Like in real Life"**. The idea is to build as many "static" .RFA file as there are machined components and nest them into a global family while applying constraints.
- The second way is what I like to call **"Generic component"**. The idea is to create generic components and switch from one supplier's brand to another by simply change the parameter values.

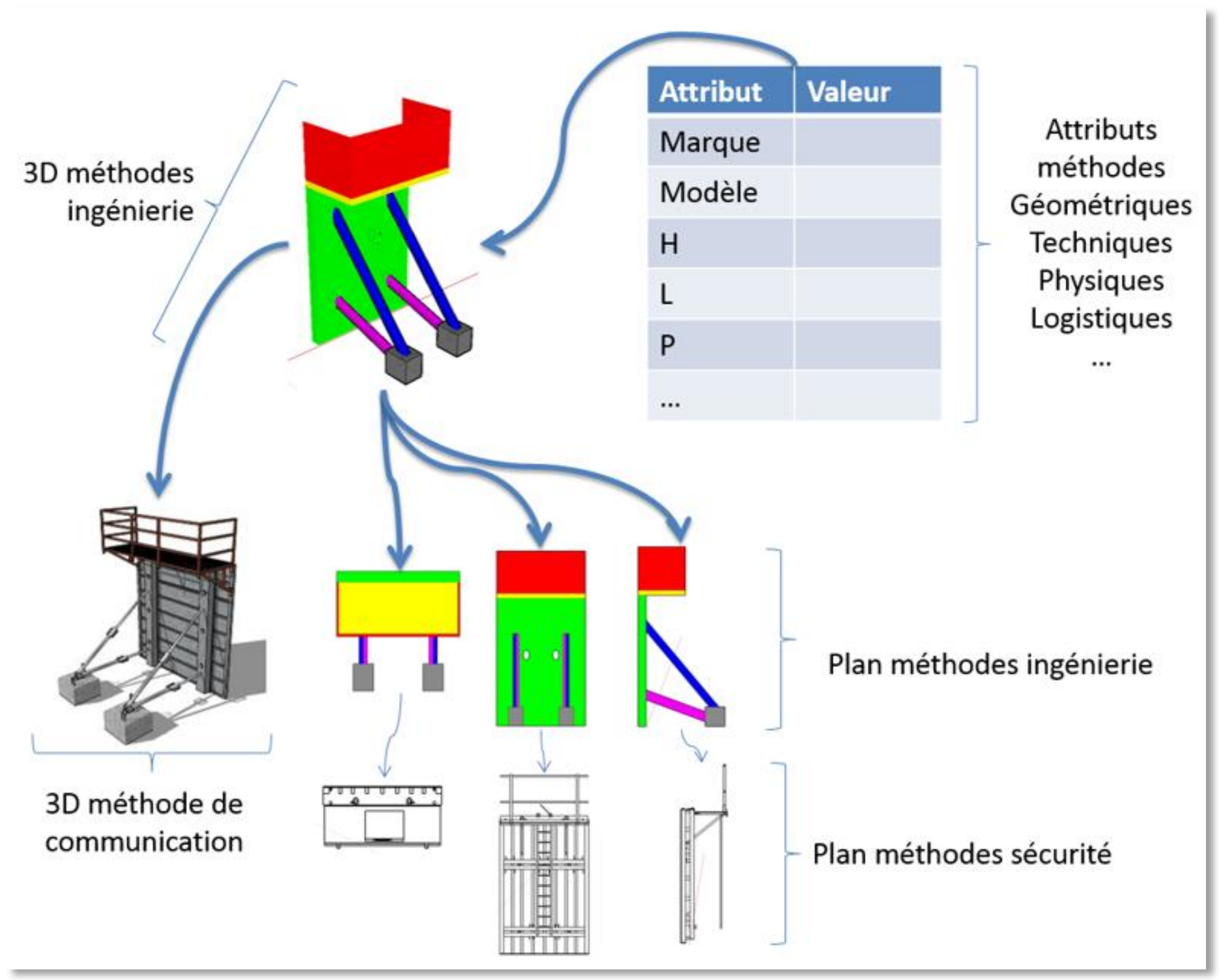


FIGURE 36 TWO WAYS OF BUILDING A FAMILY

There is no best solution but I will describe you the pros and the cons in order to make you form your own opinion:

3.2.1 Process number 1: Like in real life

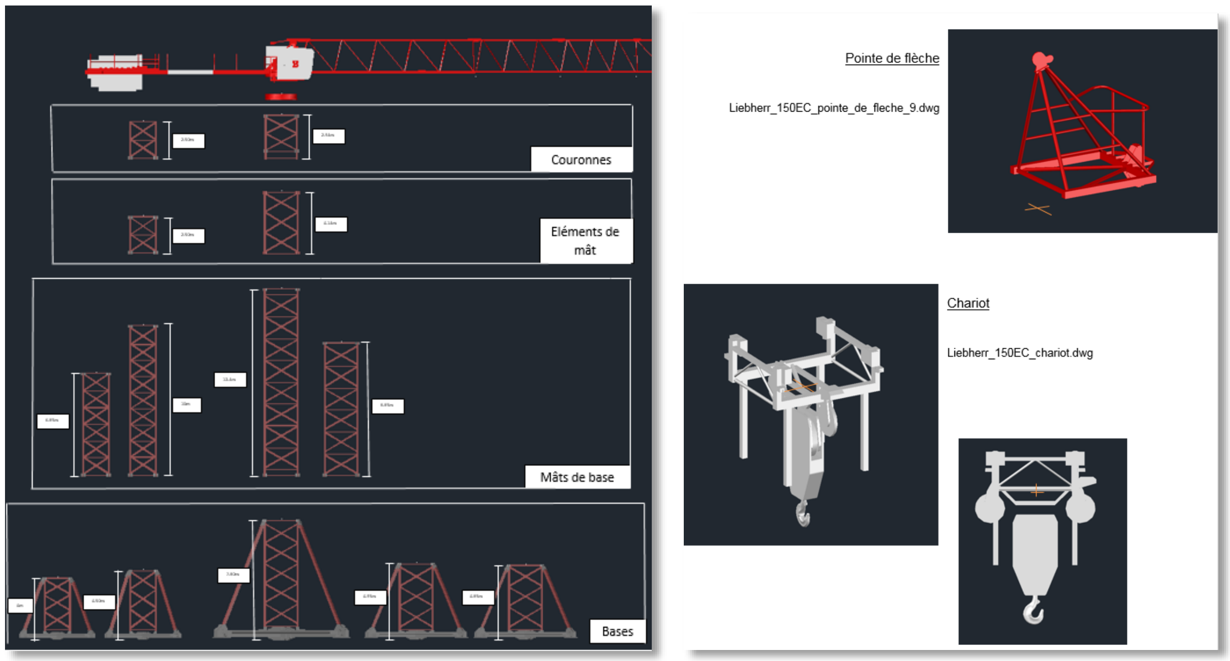


FIGURE 37 TOWER CRANE FABRICATION PARTS AS IT WILL BE ASSEMBLED ON SITE

Pros:

- If we have already a 3D DWG library, it is easier to import those geometries into nested components without having to redraw everything.
- This method allow having in your family a more detailed and realistic geometry
- It is a very intuitive way to build the family and makes it easier to reference each nested component according to the supplier codification.
- Scheduling will be based on the “shared” nested components and their internal codification so no need to add extra information parameter values based on condition or formula

Cons:

- The same equipment exists in a wide variety of suppliers so it become tedious to build and maintain lots of nested components files
- It becomes tricky when we have to decide a common nested structure for the same equipment

3.2.2 Process number 2: Generic component

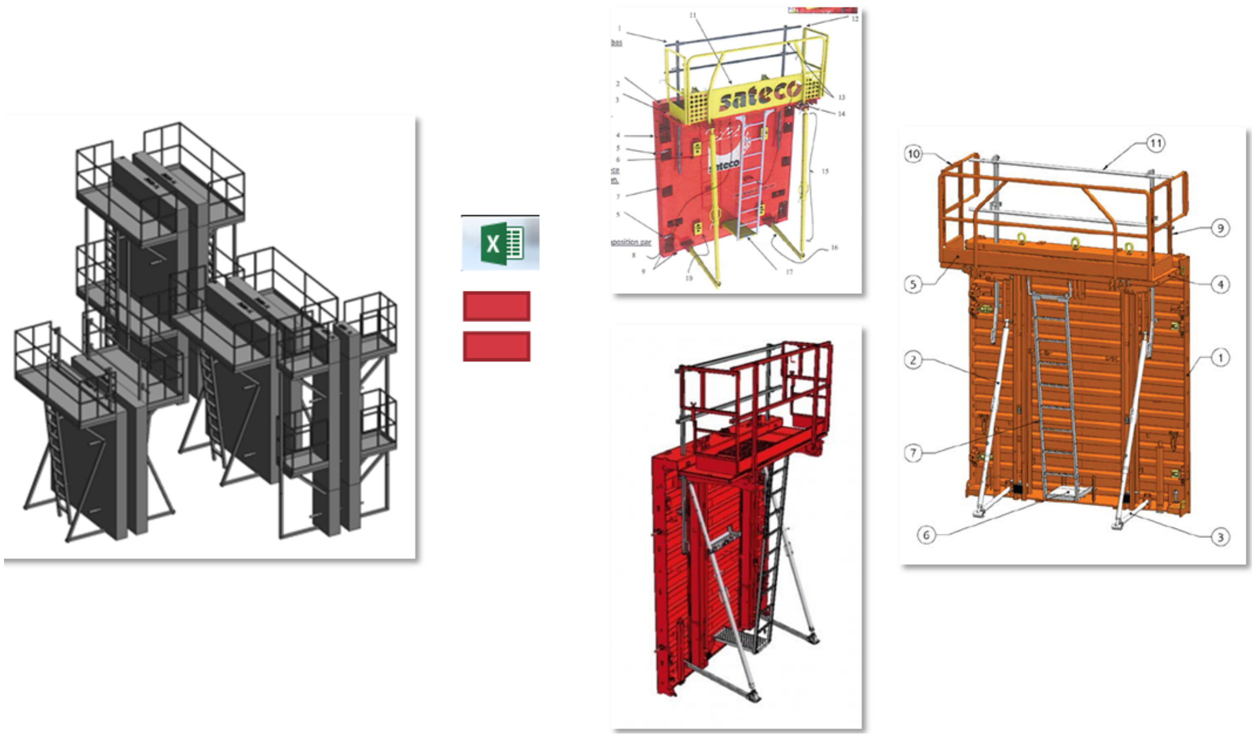


FIGURE 38 ONE WALL FORMWORK GENERIC COMPONENT WHICH DEAL WITH 3 DIFFERENT PROVIDERS

Pros:

- This method is less time consuming from the drawing point of view (less 3D details). But on the other hand it become a little bit more tricky from the parameter / constraint definitions point of view
- It makes maintenance a lot easier and the integration of a new same equipment – with the same working logic - is simply done by assigning new existing parameter values (using type catalog or Lookup Table process)
- Less 3D details involves lighter weight family files weight and increases Revit performance during the drawing process

Cons:

- This method does not deliver realistic design because geometry has to be simplified in order to be reusable (all the specific brand details, if it does not play a key role on the working site logic, are automatically removed)
- A lot more dimension parameters are involved to meet all the particular requirements
- Even though the geometry is simplified, scheduling parts still needs to be detailed: it is then necessary to add extra parameters and conditional formulas in order to take into account all the specific supplier scheduling requirements

Since Methods Engineers in Vinci do not need fabrication details but are focusing more on layout functionality and space requirements, the "Generic Component" approach has been adopted.

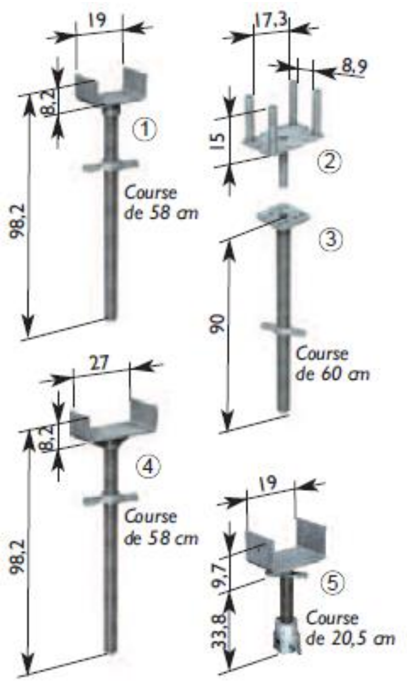


FIGURE 39 GENERIC FORK AND ALL THE DIMENSION PARAMETERS IT INVOLVES

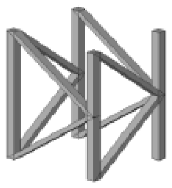
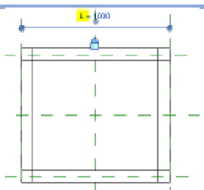
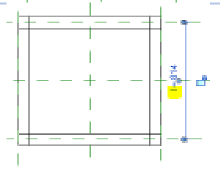
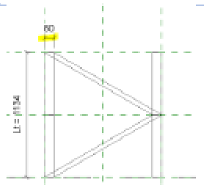
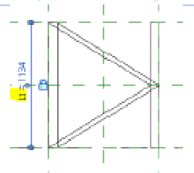
Famille	Paramètre	Type de paramètre	Illustration
	Mât_Longueur	Longueur	
	Mât_Largeur	Longueur	
	Mât_Barre_Epaisseur	Longueur	
	Mât_Treillis_Largeur	Longueur	

FIGURE 40 GENERIC TOWER CRANE MAST WITH CORRESPONDING DIMENSION PARAMETERS

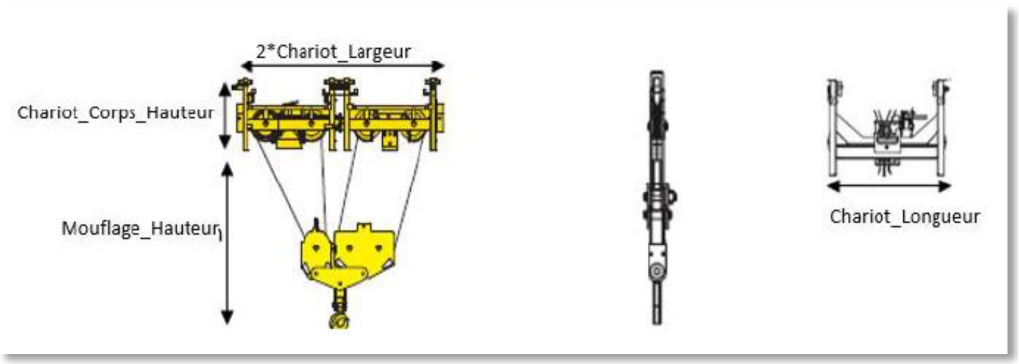


FIGURE 41 GENERIC FORK LIFT WITH CORRESPONDING DIMENSION PARAMETERS

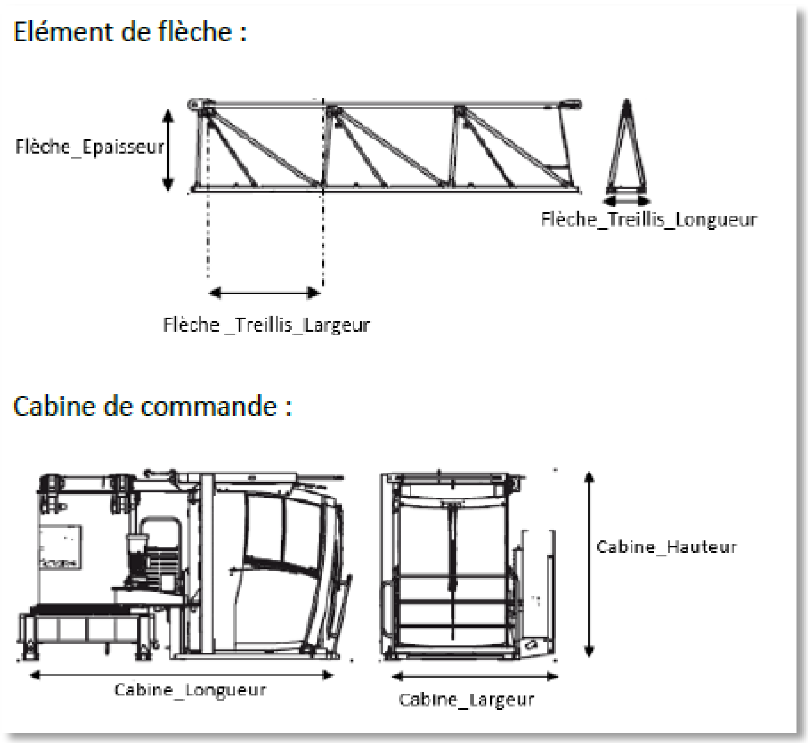


FIGURE 42 CRANE JIB ELEMENT AND CABIN DIMENSION PARAMETERS DESCRIPTION

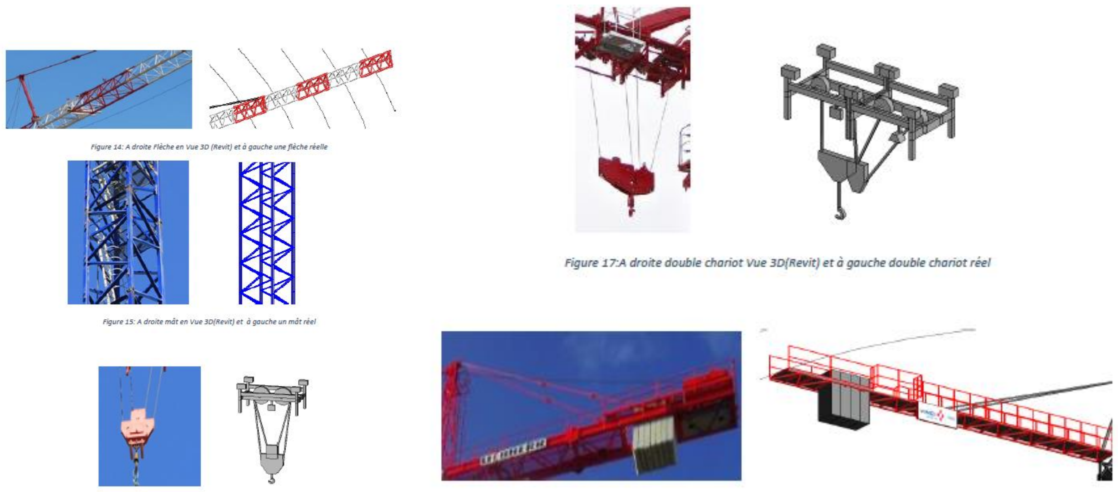
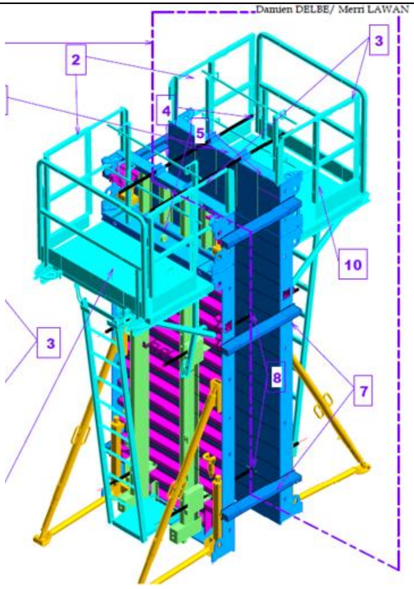
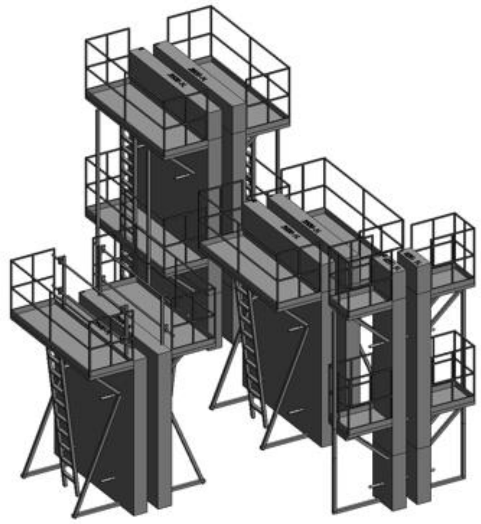
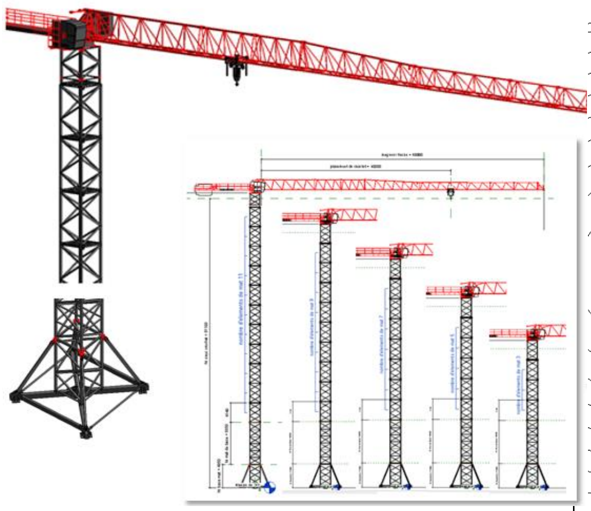
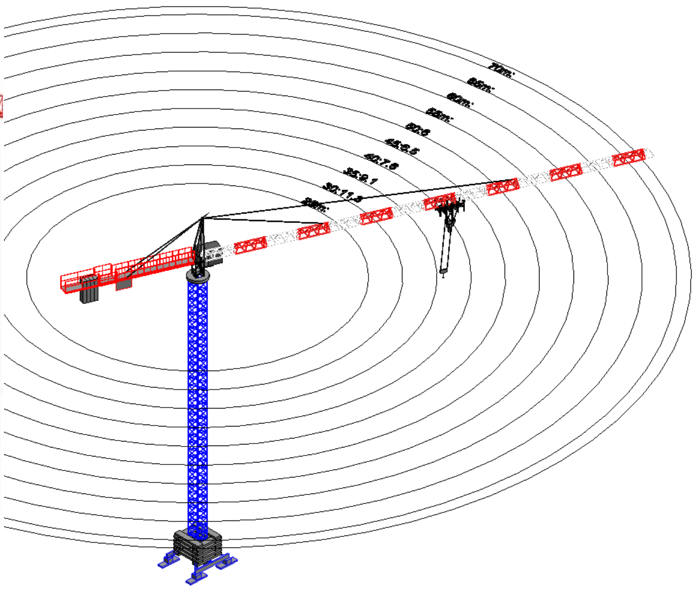
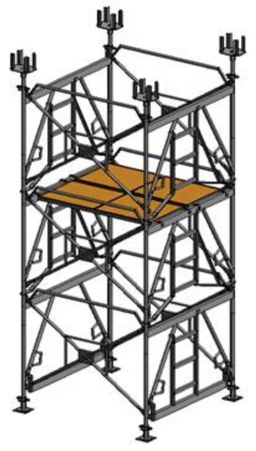
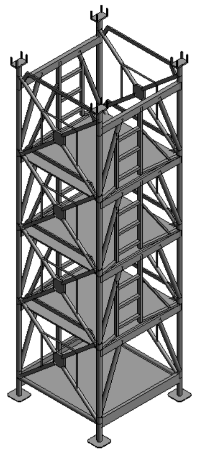


Figure 14: A droite Flèche en Vue 3D (Revit) et à gauche une flèche réelle
Figure 15: A droite mât en Vue 3D(Revit) et à gauche un mât réel
Figure 17:A droite double chariot Vue 3D(Revit) et à gauche double chariot réel

FIGURE 43 REAL LIFE ELEMENT VERSUS GENERIC ONES: NOT SO MANY DIFFERENCES!

Real life Components (fabrication needs)…	**…versus Generic Components (temporary worksite designer needs)**
 Damien DELBE/ Merri LAWAN 2 3 4 5 10 3 8 7	
	 70m. 65m. 60m. 55m. 50:6 45:6.5 40:7.6 35:9.1 30:11.5 25m.

4 Pushing the Revit family limits with the API

4.1 The added value of API

During the implementation process in our company, even though family features are great, we deals with some limitations. Our API understanding helps us to find workaround in order to achieve our objective:

- **Data amount:** it can be very tricky when we have to deal with a lot of data inside the same equipment. We can still use Lookup Table feature but we are limited to dimension parameters (we cannot call string parameter for example) and if we change the data we have to reload the .csv file into each family which is time consuming. Last but not least, Revit restricts to users the data structure and file format and these files cannot be shared on a secured server with user rights or on a public website.
- **Formula complexity:** in some cases formulas are hard to implement - lots of conditional statements, unit restriction -, hard to read, and manual typing can be error-prone. Programing helps organize in a more friendly way calculations logic.
- **Maintainability across CAD Solutions:** in computer sciences it is always better to separate programming layers: it is called “separation of concerns”. This is possible with Revit API because it is using the Microsoft.NET platform and an object oriented language (C# or VB.NET). “Separation of concerns” in programming distinguish 3 layers:
  - Data layer: it describes how we retrieve the data from a storage file like a .CSV or .XLSX file
  - Business layer: it defines business objects – with its properties and methods - and describes how business objects interact with data. In our case, a business object could be a crane or a wall formwork definition (in fact, it is the same way Revit API define building components objects).

- <u>User interface layer:</u> it describes - thought graphical user interface controls – the interaction between the user input and the business layer.

This 3 layers concept enables us to maintain for each custom equipment a common database and business layer without being software dependent. We only have to deal with specific user interface layer for each CAD software because the interaction with the component is software dependent.

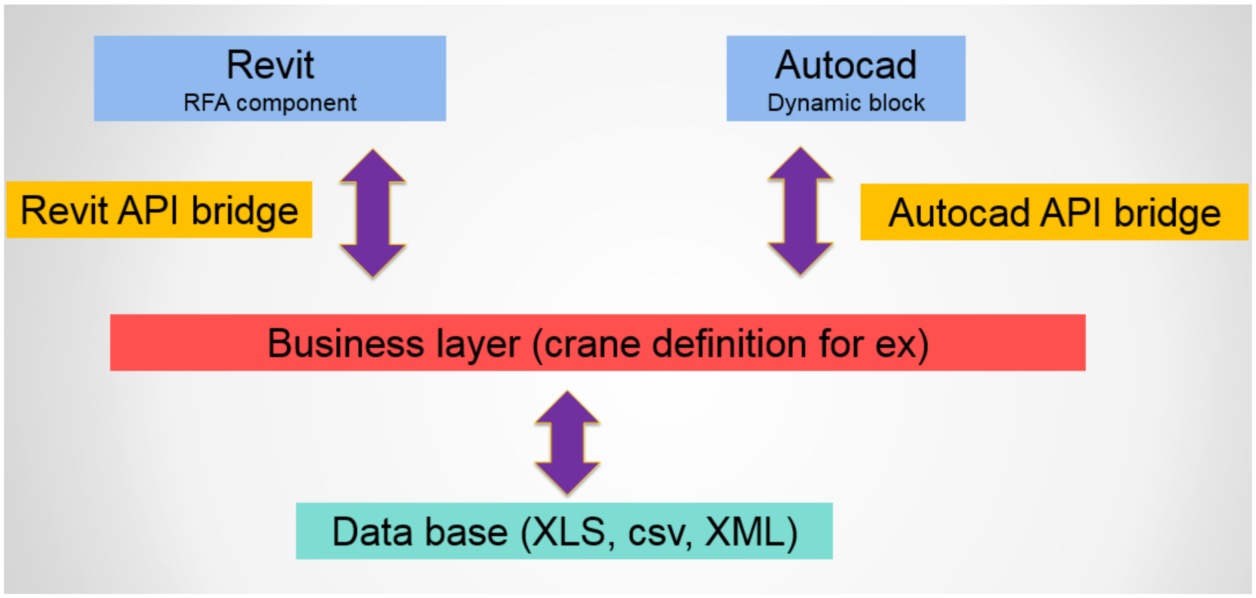


FIGURE 44 SEPARATION OF CONCERNS AND CODE REUSE BETWEEN CAD SOFTWARE

- **Graphical User Controls:** in the Revit API, we can benefit from the great Visual Studio's graphical controls (Windows Form, WPF) in order to build a more user friendly dialog box than the default family property panel in Revit. For example we can:
  - Create combo box item values onto instance parameters while restricting choices. The only workaround to do this without the API is to use Family Type parameter and nest family types into the final family. This increases the complexity of the family creation and has one limitation: all the nested types of the same category will appear on the final list even if they are not related.
  - Separate into different tabs the decision stages: this helps reduce the overall user interface and guides the user step by step
  - Allow chain reaction between parameter values through event control: very useful when you have multiple cascading choices
  - Easily restrict user choices by fixing minimum/maximum values or step increment on graphical controls without using any if statement.

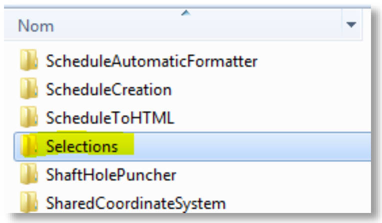
4.2 API useful functionalities and .NET framework tools

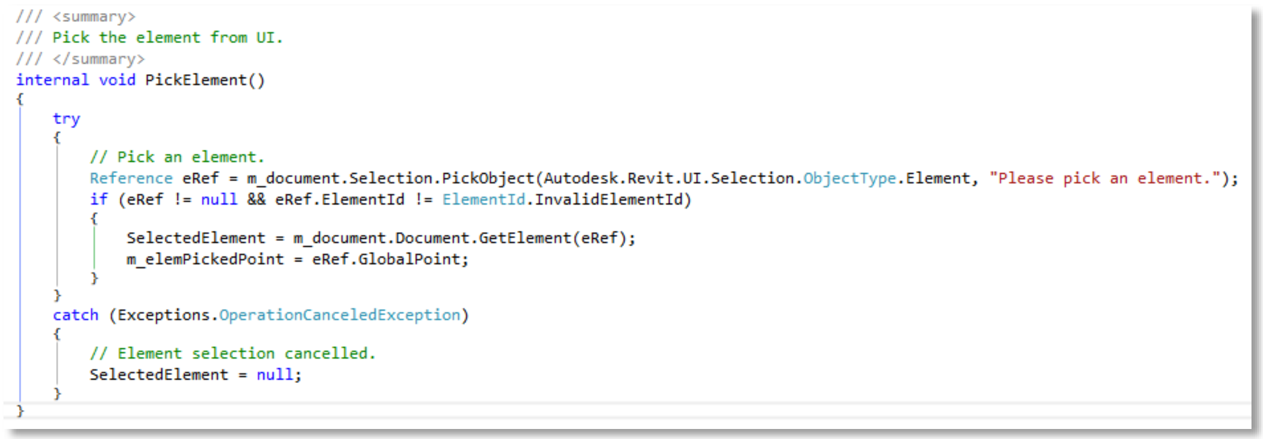
During this chapter, we will see different features I am using in order to achieve an API "bridge" between the user interface (.NET graphical controls) and the Revit families. Then we will see some useful features that .NET framework provides in order to create great both user interface and data base connection.

4.2.1 Revit API useful features

1. How to deal with Revit family selection

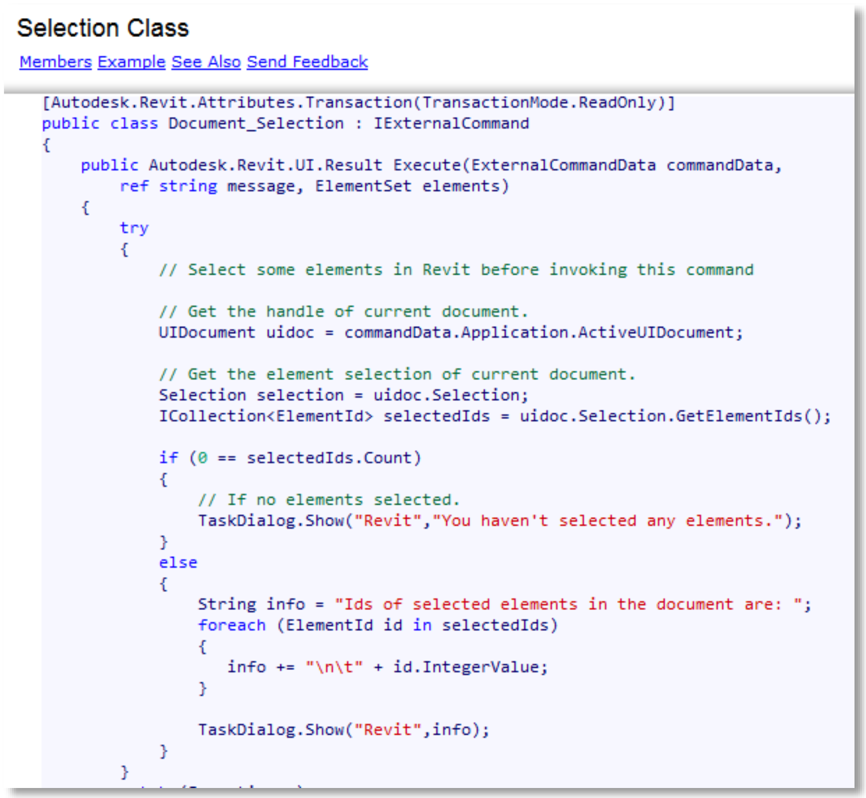
Some sample code show how to prompt the user to pick an Element in Revit and specify on which category to filter:





```
/// <summary>
/// Pick the element from UI.
/// </summary>
internal void PickElement()
{
    try
    {
        // Pick an element.
        Reference eRef = m_document.Selection.PickObject(Autodesk.Revit.UI.Selection.ObjectType.Element, "Please pick an element.");
        if (eRef != null && eRef.ElementId != ElementId.InvalidElementId)
        {
            SelectedElement = m_document.Document.GetElement(eRef);
            m_elemPickedPoint = eRef.GlobalPoint;
        }
    }
    catch (Exceptions.OperationCanceledException)
    {
        // Element selection cancelled.
        SelectedElement = null;
    }
}
```

It also describes how to retrieve the user current selection in the project:

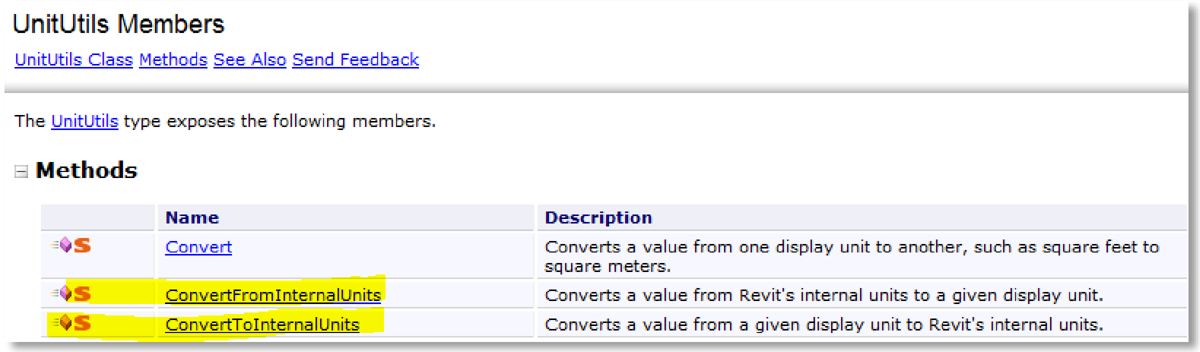

Selection Class

Members Example See Also Send Feedback

```
[Autodesk.Revit.Attributes.Transaction(TransactionMode.ReadOnly)]
public class Document_Selection : IExternalCommand
{
    public Autodesk.Revit.UI.Result Execute(ExternalCommandData commandData,
        ref string message, ElementSet elements)
    {
        try
        {
            // Select some elements in Revit before invoking this command

            // Get the handle of current document.
            UIDocument uidoc = commandData.Application.ActiveUIDocument;

            // Get the element selection of current document.
            Selection selection = uidoc.Selection;
            ICollection<ElementId> selectedIds = uidoc.Selection.GetElementIds();

            if (0 == selectedIds.Count)
            {
                // If no elements selected.
                TaskDialog.Show("Revit","You haven't selected any elements.");
            }
            else
            {
                String info = "Ids of selected elements in the document are: ";
                foreach (ElementId id in selectedIds)
                {
                   info += "\n\t" + id.IntegerValue;
                }

                TaskDialog.Show("Revit",info);
            }
        }
```

2. Deal with Revit unit

In order to deal with unit conversion without knowing the current project unit, the Revit API provides a very useful class called “UnitUtils”. For example if you always play with centimeters unit in your custom UI, you just have to specify it by passing it as an argument when you get / set dimension parameter values from / to Revit.


UnitUtils Members

UnitUtils Class Methods See Also Send Feedback

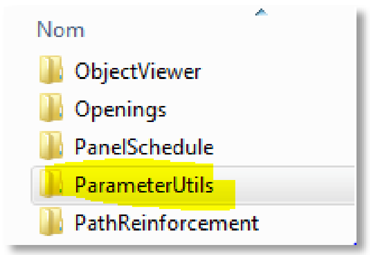
The UnitUtils type exposes the following members.

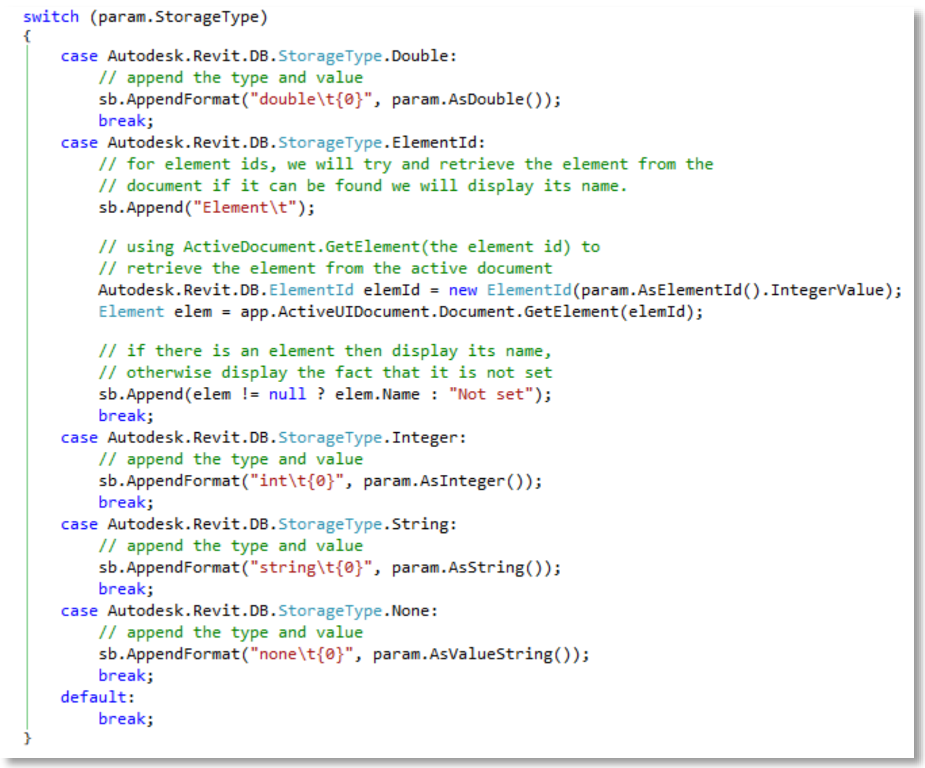
Methods

	Name	Description
S	Convert	Converts a value from one display unit to another, such as square feet to square meters.
S	ConvertFromInternalUnits	Converts a value from Revit's internal units to a given display unit.
S	ConvertToInternalUnits	Converts a value from a given display unit to Revit's internal units.

3. Retrieve or set values both type or instance parameter

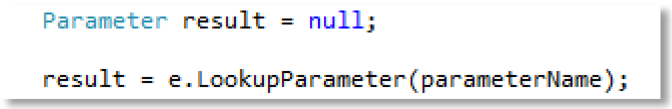
In the sample SDK folder there is a great piece of code called “ParameterUtils” which describes how to get/set parameter information base on its Storage Type (Volume, Text, ElementId, length):





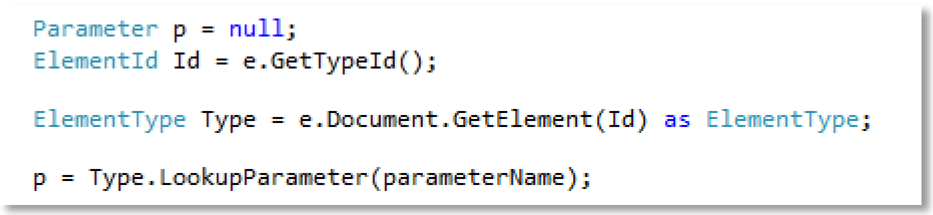
```
switch (param.StorageType)
{
    case Autodesk.Revit.DB.StorageType.Double:
        // append the type and value
        sb.AppendFormat("double\t{0}", param.AsDouble());
        break;
    case Autodesk.Revit.DB.StorageType.ElementId:
        // for element ids, we will try and retrieve the element from the
        // document if it can be found we will display its name.
        sb.Append("Element\t");

        // using ActiveDocument.GetElement(the element id) to
        // retrieve the element from the active document
        Autodesk.Revit.DB.ElementId elemId = new ElementId(param.AsElementId().IntegerValue);
        Element elem = app.ActiveUIDocument.Document.GetElement(elemId);

        // if there is an element then display its name,
        // otherwise display the fact that it is not set
        sb.Append(elem != null ? elem.Name : "Not set");
        break;
    case Autodesk.Revit.DB.StorageType.Integer:
        // append the type and value
        sb.AppendFormat("int\t{0}", param.AsInteger());
        break;
    case Autodesk.Revit.DB.StorageType.String:
        // append the type and value
        sb.AppendFormat("string\t{0}", param.AsString());
        break;
    case Autodesk.Revit.DB.StorageType.None:
        // append the type and value
        sb.AppendFormat("none\t{0}", param.AsValueString());
        break;
    default:
        break;
}
```

To retrieve an instance parameter, this is very straightforward:



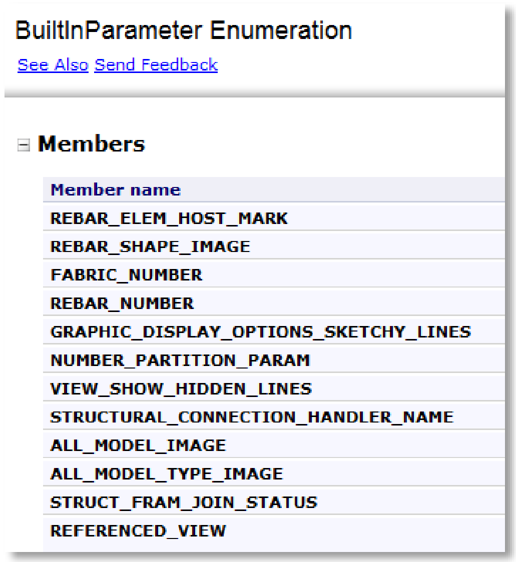
```
Parameter result = null;

result = e.LookupParameter(parameterName);
```

To retrieve a type parameter on an Element, you have to write one more line of code:



```
Parameter p = null;
ElementId Id = e.GetTypeId();

ElementType Type = e.Document.GetElement(Id) as ElementType;

p = Type.LookupParameter(parameterName);
```

As much as possible, when you call Revit internal parameters (for example wall length, volume or surface), it is best to use **BuiltInParameter** enumerations so it will not be language dependent (it means your plugin will work both on English and French Revit version):

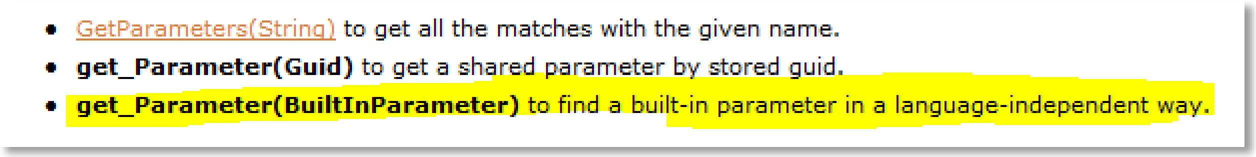


BuiltInParameter Enumeration

See Also Send Feedback

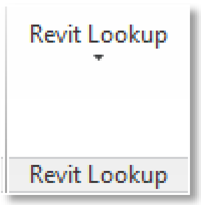
⊟ **Members**

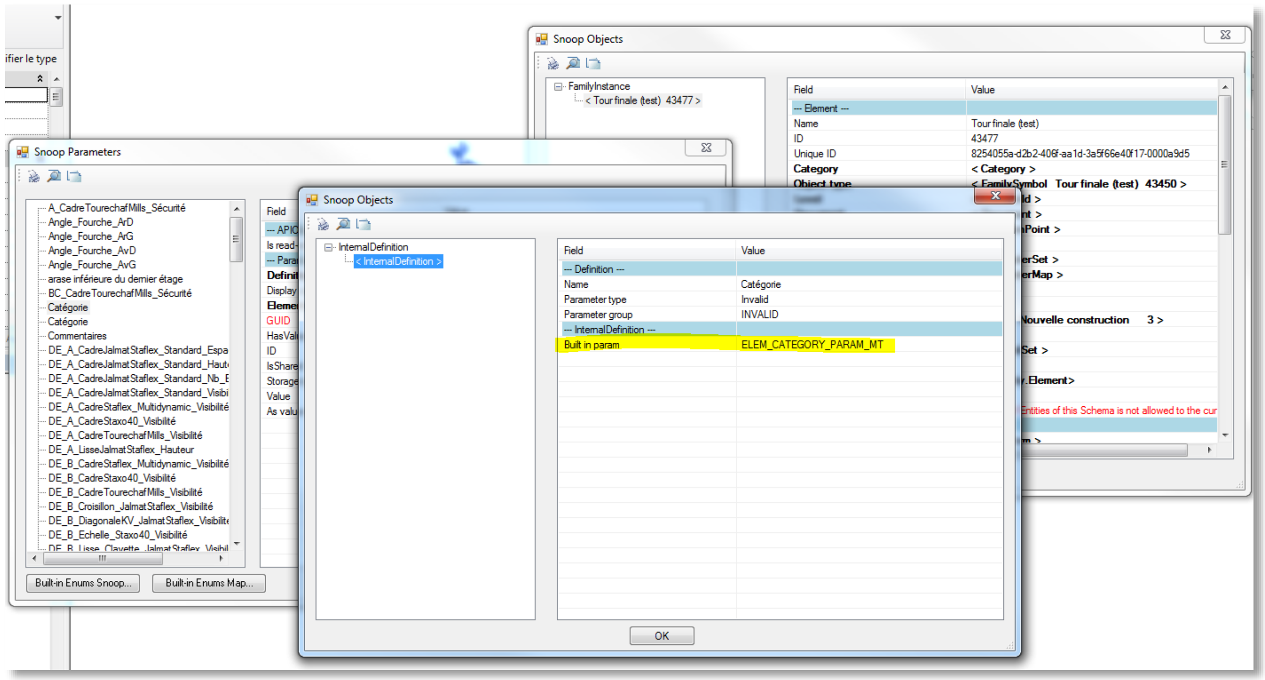
Member name
REBAR_ELEM_HOST_MARK
REBAR_SHAPE_IMAGE
FABRIC_NUMBER
REBAR_NUMBER
GRAPHIC_DISPLAY_OPTIONS_SKETCHY_LINES
NUMBER_PARTITION_PARAM
VIEW_SHOW_HIDDEN_LINES
STRUCTURAL_CONNECTION_HANDLER_NAME
ALL_MODEL_IMAGE
ALL_MODEL_TYPE_IMAGE
STRUCT_FRAM_JOIN_STATUS
REFERENCED_VIEW



- GetParameters(String) to get all the matches with the given name.
- **get_Parameter(Guid)** to get a shared parameter by stored guid.
- **get_Parameter(BuiltInParameter)** to find a built-in parameter in a language-independent way.

And if you do not know which enumeration matches your parameter name, you can simply use Revit Lookup plugin which is freely provided on the following GitHub link: https://github.com/jeremytammik/RevitLookup/releases/tag/2016.0.0.6 :





Snoop Objects

FamilyInstance
< Tour finale (test) 43477 >

Field	Value
--- Element ---	
Name	Tour finale (test)
ID	43477
Unique ID	8254055a-d2b2-406f-aa1d-3a5f66e40f17-0000a9d5
Category	< Category >
Object type	< FamilySymbol Tour finale (test) 43450 >

Snoop Parameters

A_CadreTourechafMills_Sécurité
Angle_Fourche_ArD
Angle_Fourche_ArG
Angle_Fourche_AvD
Angle_Fourche_AvG
arase inférieure du dernier étage
BC_CadreTourechafMills_Sécurité
Catégorie
Catégorie
Commentaires
DE_A_CadreJalmatStaflex_Standard_Espa
DE_A_CadreJalmatStaflex_Standard_Haut
DE_A_CadreJalmatStaflex_Standard_Nb_E
DE_A_CadreJalmatStaflex_Standard_Visibi
DE_A_CadreStaflex_Multidynamic_Visibilité
DE_A_CadreStaxo40_Visibilité
DE_A_CadreTourechafMills_Visibilité
DE_A_LisseJalmatStaflex_Hauteur
DE_B_CadreStaflex_Multidynamic_Visibilité
DE_B_CadreStaxo40_Visibilité
DE_B_CadreTourechafMills_Visibilité
DE_B_Croisillon_JalmatStaflex_Visibilité
DE_B_DiagonaleKV_JalmatStaflex_Visibilite
DE_B_Echelle_Staxo40_Visibilité

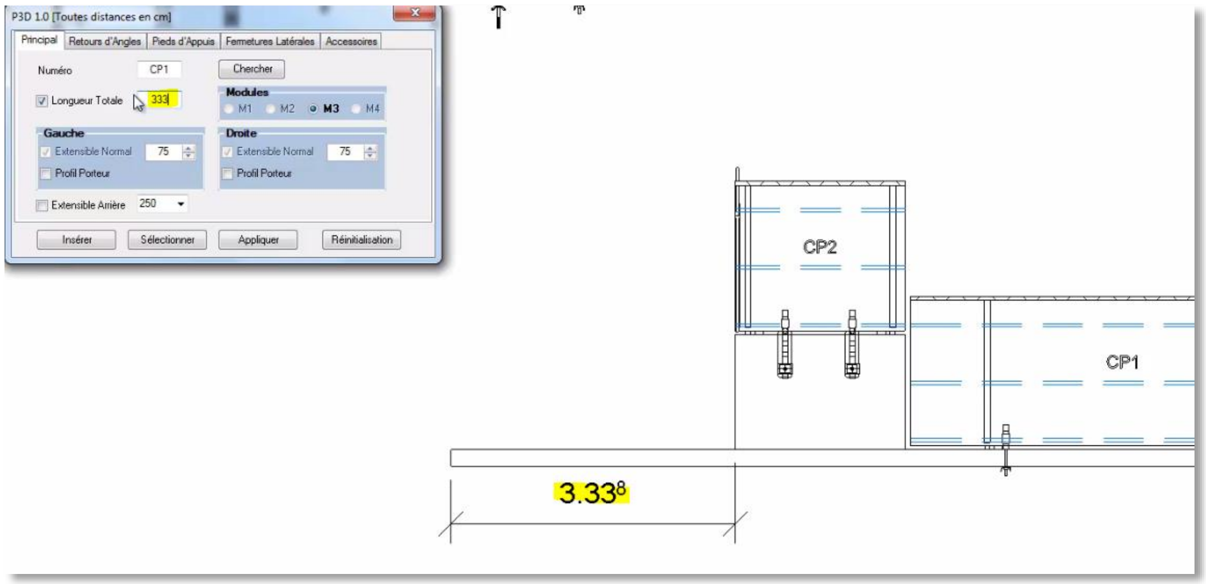
Snoop Objects

InternalDefinition
< InternalDefinition >

Field	Value
--- Definition ---	
Name	Catégorie
Parameter type	Invalid
Parameter group	INVALID
--- InternalDefinition ---	
Built in param	ELEM_CATEGORY_PARAM_MT

4. **Displaying modeless form**

In most cases, it is very convenient to choose equipment family options on a windows form and keep it open while working on the project: these forms are called Modeless Form (as opposed to modal dialog which blocks the application flow until information required to continue is entered). It allows us to see both equipment options and the project context (if you want to measure the wall length while choosing safety platform dimensions for example):


P3D 1.0 [Toutes distances en cm]

Numéro: CP1

[✓] Longueur Totale: 333

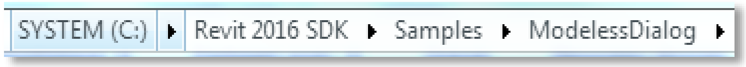
Modules: M1 M2 [✓ M3] M4

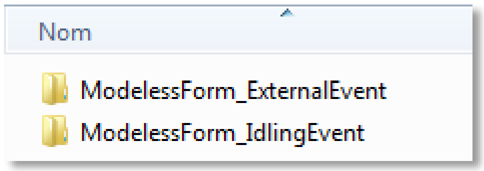
Gauche: [✓] Extensible Normal 75; [ ] Profil Porteur

Droite: [✓] Extensible Normal 75; [ ] Profil Porteur

[ ] Extensible Arrière 250

Unfortunately, Revit API does not allow us to make use of transaction outside Revit external command context, in other words you cannot write parameters values while keeping your windows form open). But there is a workaround in order to achieve this result: this is explained on Jérémy Tammik blog "the building coder". He suggests to use **Idling Event** or **External Event** features. If it does not sounds familiar to you, he kindly implemented two ready-to-use templates in the Revit SDK Sample Folder:


SYSTEM (C:) ▸ Revit 2016 SDK ▸ Samples ▸ ModelessDialog ▸

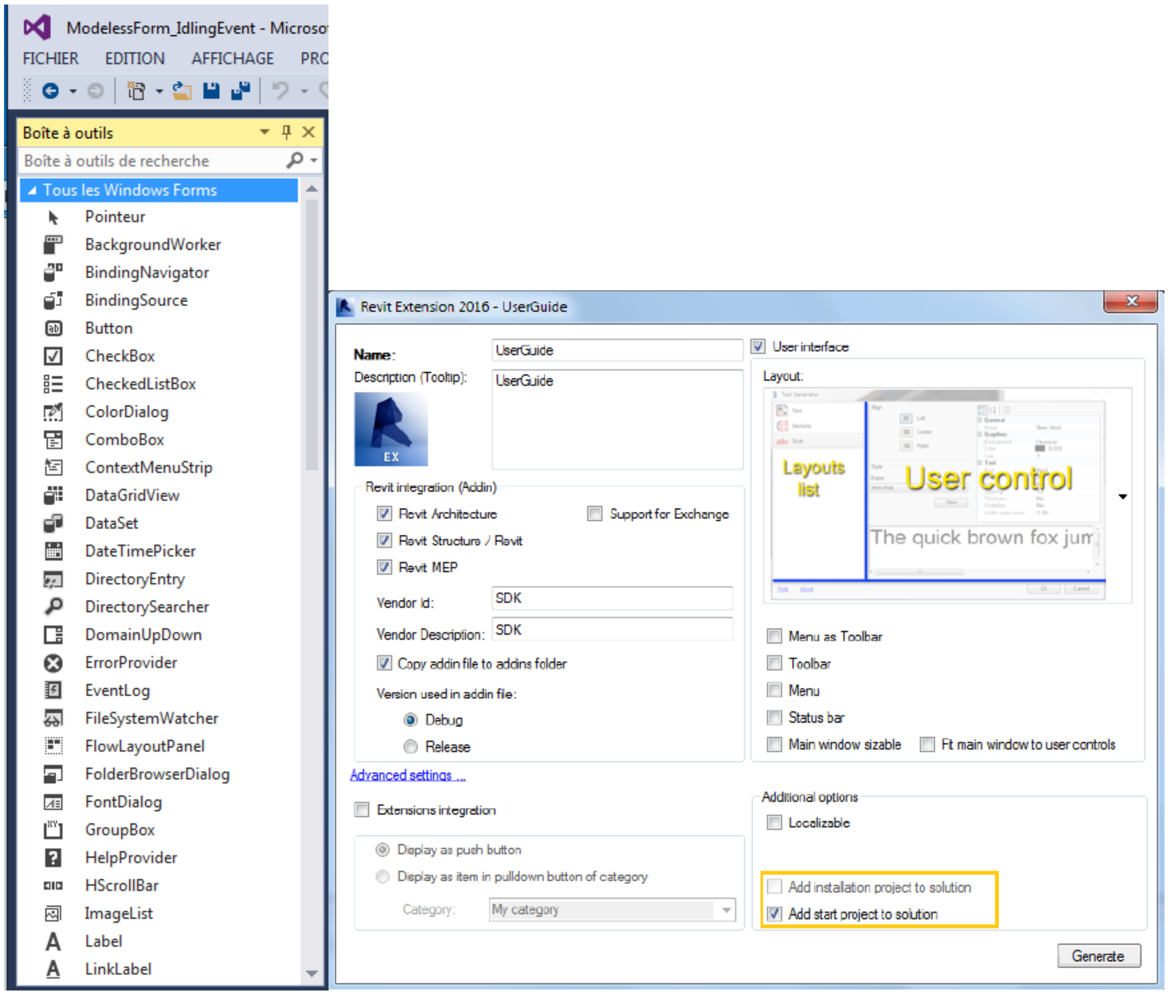

Nom

ModelessForm_ExternalEvent

ModelessForm_IdlingEvent

4.2.2 *Useful .NET Tools*

1. **Make use of powerful graphical controls**



Name: UserGuide
Description (Tooltip): UserGuide

Revit integration (Addin)
[✓] Revit Architecture
[ ] Support for Exchange
[✓] Revit Structure / Revit
[✓] Revit MEP
Vendor id: SDK
Vendor Description: SDK
[✓] Copy addin file to addins folder
Version used in addin file: [✓ Debug] Release

[ ] Extensions integration
[✓ Display as push button] Display as item in pulldown button of category
Category: My category

[✓] User interface
Layout:
[ ] Menu as Toolbar
[ ] Toolbar
[ ] Menu
[ ] Status bar
[ ] Main window sizable
[ ] Fit main window to user controls

Additional options
[ ] Localizable
[ ] Add installation project to solution
[✓] Add start project to solution

FIGURE 45 VISUAL STUDIO ENVIRONMENT AND ITS NUMEROUS GRAPHICAL CONTROLS

2. Create business oriented objects

Programming with Revit API allow you to take advantage of object oriented language features. That means that you can create specific business oriented classes like crane, wall formwork with their own properties and methods (methods can be specific Excel data querying, or read/write parameters values in Revit). Objects oriented language also allow us to create very generic classes and inherit more specific classes while keeping parent class properties and methods (for example we can inherit tower crane from a generic lifting equipment class). You can also (like the nesting feature in the family editor) nest classes into other classes (for example a tower crane class could contain: a fork lift class, a ballast class, a load curve class, etc.) in order to break down complex object structure into simpler objects.

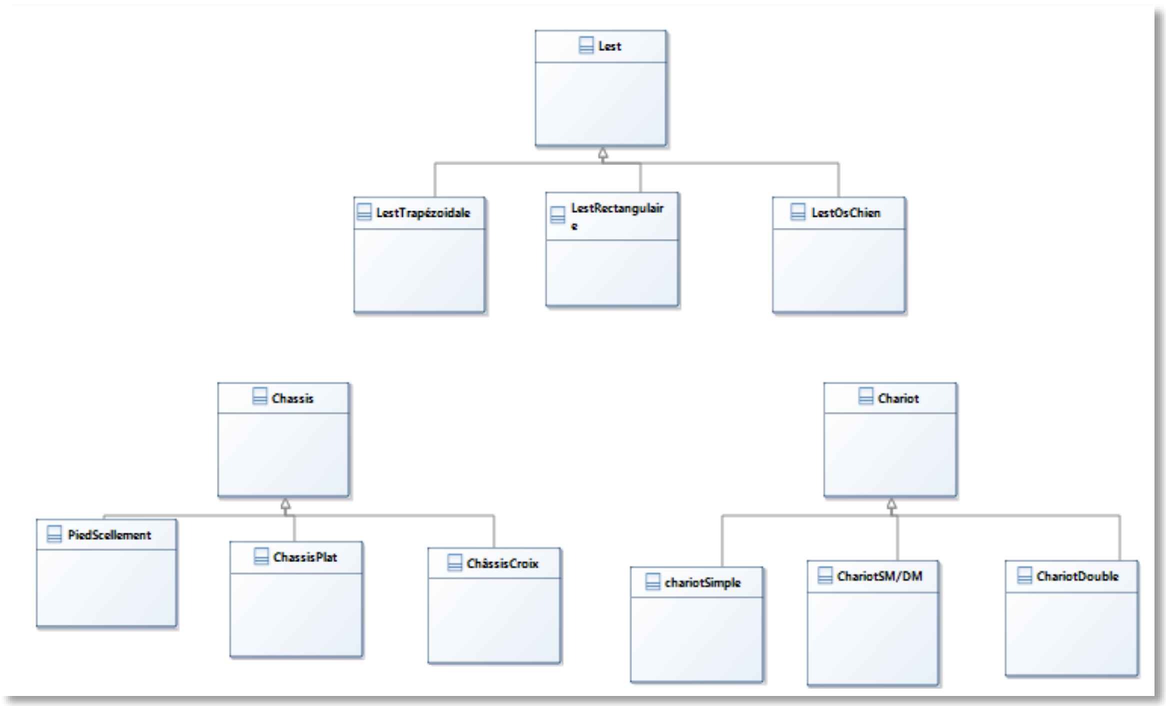
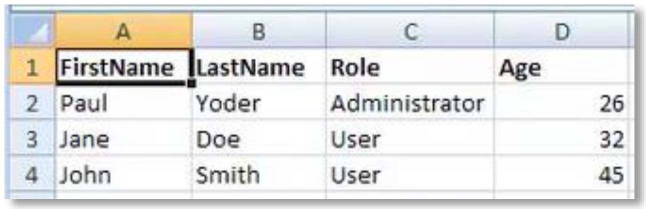


FIGURE 46 A TOWER CRANE CLASS DIAGRAM EXAMPLE

3. **Interact easily with database**
   You can use the .NET framework API in order to interact with Office softwares (Excel, Access) or SQL Data Base. Use **Linq (Language Integrated Query)** features to quickly and easily query object collections in a human readable language:

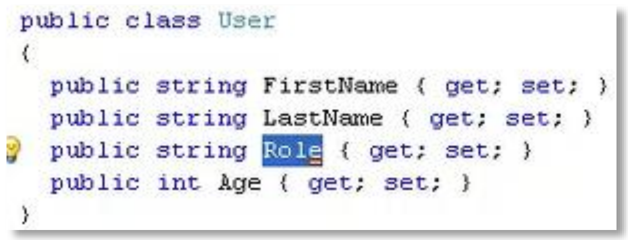
```
IEnumerable<string> result =
  from c in myCustomers
  where c.Name.StartsWith("B")
  select c.Name;
```

   You can also use additional free open source libraries in order to query and fill in data object collection from Excel with fewer lines of code. Good example I found are **LinqtoExcel**, **MoreLinq**, or **EEPlus.** One interesting feature in **LinqToExcel** is to create object collections by matching Excel column header name with the corresponding object property.



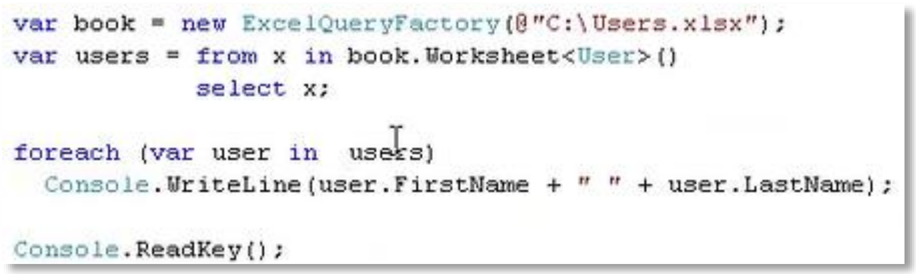
	A	B	C	D
1	FirstName	LastName	Role	Age
2	Paul	Yoder	Administrator	26
3	Jane	Doe	User	32
4	John	Smith	User	45

FIGURE 47 EXCEL DATA WITH NAMED COLUMN HEADER



```
public class User
{
  public string FirstName { get; set; }
  public string LastName { get; set; }
  public string Role { get; set; }
  public int Age { get; set; }
}
```

FIGURE 48 OBJECT DEFINITION WITH NAMED PROPERTIES THAT MATCH EXCEL NAMES



```
var book = new ExcelQueryFactory(@"C:\Users.xlsx");
var users = from x in book.Worksheet<User>()
            select x;

foreach (var user in  users)
  Console.WriteLine(user.FirstName + " " + user.LastName);

Console.ReadKey();
```

FIGURE 49 SIMPLE QUERY WHICH FILL IN THE OBJECT COLLECTION

5 Real case study in VINCI CONSTRUCTION FRANCE

During this chapter I will show you different usages of the Revit family and API features in order to make our custom equipments a little bit more “clever” than we are used to in AutoCAD. I hope it will give you some good insight when you will come back to your offices.

5.1 Tower crane

Global requirements specification for tower cranes are:

- A helping user decision interface: because there are a lot of criteria to consider when choosing a crane, the interface must guide step by step the user.

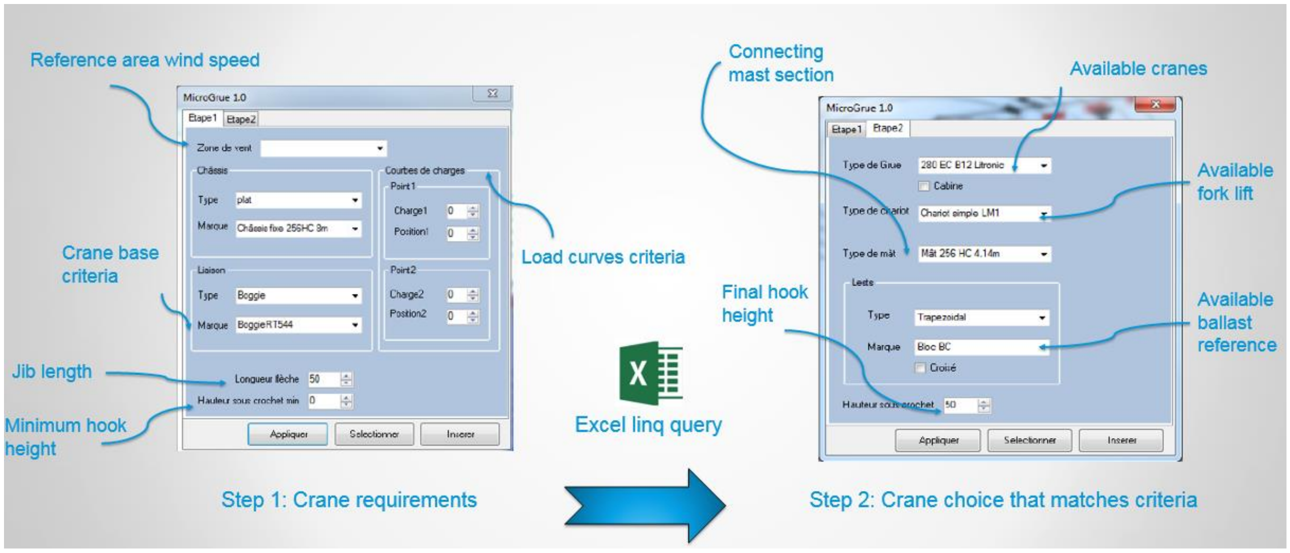
- Once the crane is chosen, when inserted into Revit, the drawing must show all the relevant and up to date information for the application dossier (mayor's office authorization)
- Easy data storage and maintenance
- Easy family maintenance by creating generic components

1. Helping decision interface:

We decided to create a user interface organized in 2 tabs:

- The first tab is regrouping the main criteria which help Methods Engineers filter the cranes that matches:
  - Reference area wind speed (this restricts the maximum hook height)
  - Crane base options based on its space requirement and load dimensioning
  - Jib length by measuring in the Revit drawing the building footprint
  - Minimum hook height also by measuring in Revit elevation view the maximum building height covered by the crane
- The second tab allow the user to choose the available options among the crane configurations that matches all the previous criteria:
  - Fork lift option (final fork lift speed and loading capabilities will depend on it)
  - Ballast reference option and connecting mast section (important to check spacing requirement on the field)
  - Final hook height

A search button is doing an Excel LINQ query with the first tab criteria and show the result in the second tab:



MicroGrue 1.0 — Etape1

Zone de vent:

Châssis — Type: plat; Marque: Châssis fixe 256HC 9m

Liaison — Type: Boggie; Marque: BoggieRT544

Courbes de charges — Point1: Charge1 0, Position1 0; Point2: Charge2 0, Position2 0

Longueur flèche: 50

Hauteur sous crochet min: 0

MicroGrue 1.0 — Etape2

Type de Grue: 280 EC B12 Litronic

Cabine

Type de chariot: Chariot simple LM1

Type de mât: Mât 256 HC 4.14m

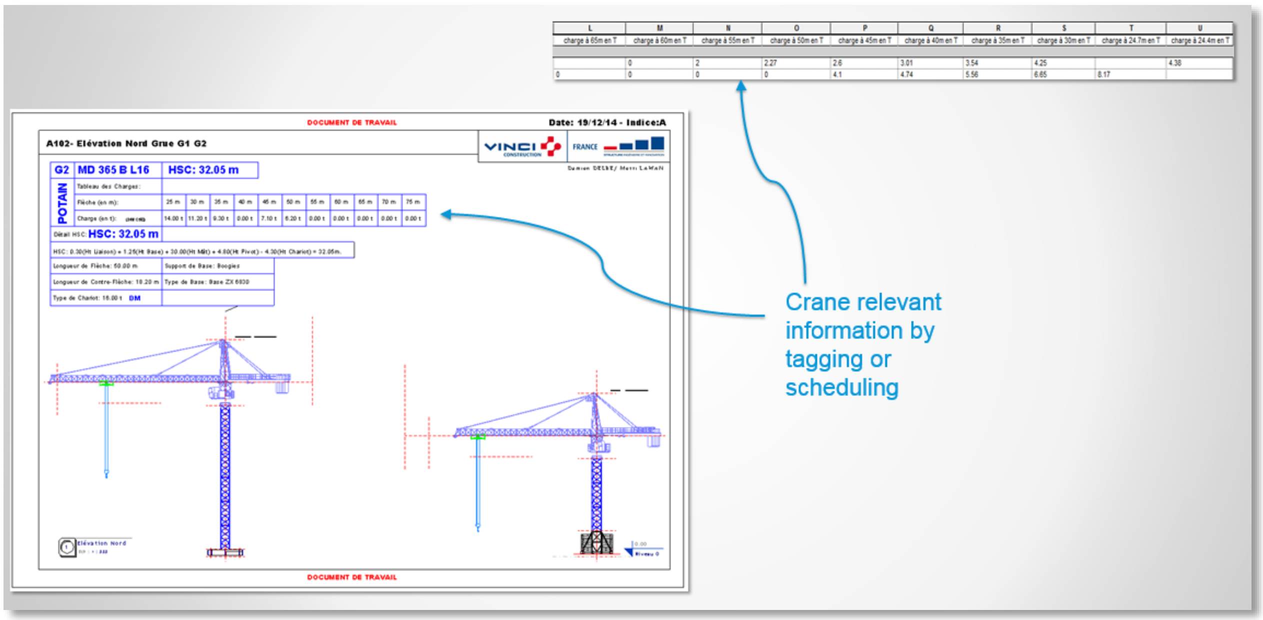
Lests — Type: Trapezoidal; Marque: Bloc BC

Croisé

Hauteur sous crochet: 50

2. Drawing information

- In order to have up to date information in the Revit drawing, we created a multiple field tag which contains the crane characteristics and load curves:



L	M	N	O	P	Q	R	S	T	U
charge à 65m en T	charge à 60m en T	charge à 55m en T	charge à 50m en T	charge à 45m en T	charge à 40m en T	charge à 35m en T	charge à 30m en T	charge à 24.7m en T	charge à 24.4m en T
	0	2	2.27	2.6	3.01	3.54	4.25		4.38
0	0	0	0	4.1	4.74	5.56	6.65	8.17	

DOCUMENT DE TRAVAIL

Date: 19/12/14 - Indice:A

A102- Elévation Nord Grue G1 G2

Damien DELBE/ Merri LAWAN

G2	MD 365 B L16	HSC: 32.05 m									
POTAIN	Tableau des Charges:										
	Flèche (en m):	25 m	30 m	35 m	40 m	45 m	50 m	55 m	60 m	65 m	70 m	75 m
	Charge (en t):	14.00 t	11.20 t	9.30 t	0.00 t	7.10 t	6.20 t	0.00 t	0.00 t	0.00 t	0.00 t	0.00 t

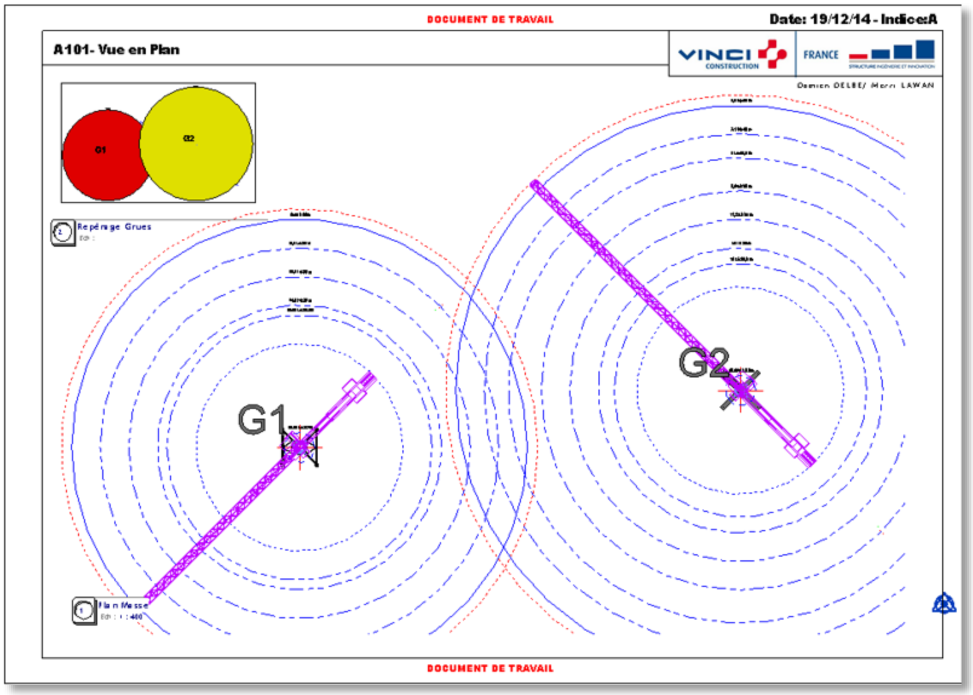
Détail HSC: HSC: 32.05 m

HSC: 0.30(Ht Liaison) + 1.25(Ht Base) + 30.00(Ht Mât) + 4.80(Ht Pivot) - 4.30(Ht Chariot) = 32.05m.

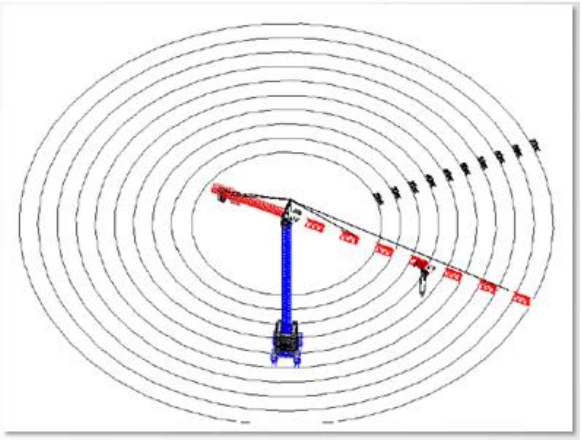
Longueur de Flèche: 60.00 m	Support de Base: Boogies
Longueur de Contre-Flèche: 18.20 m	Type de Base: Base ZX 6030
Type de Chariot: 16.00 t DM	

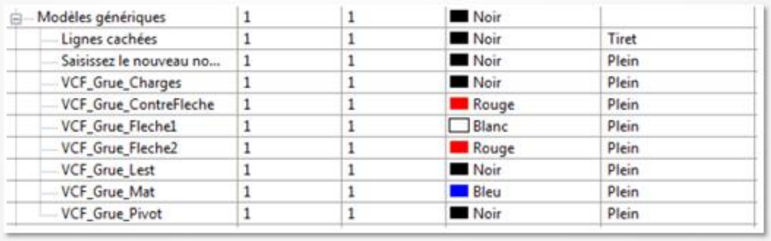
DOCUMENT DE TRAVAIL

- By using symbolic circles in 2D plan view, we clarify the crane interferences and indicate which one is above the over:



- Graphical charter is also a big deal: due to a communication issue, the site installation layout must reflect the identity of VINCI CONSTRUCTION FRANCE. That's why we use object styles features in Revit which enable us personalize color in the Revit Project without touching the family:

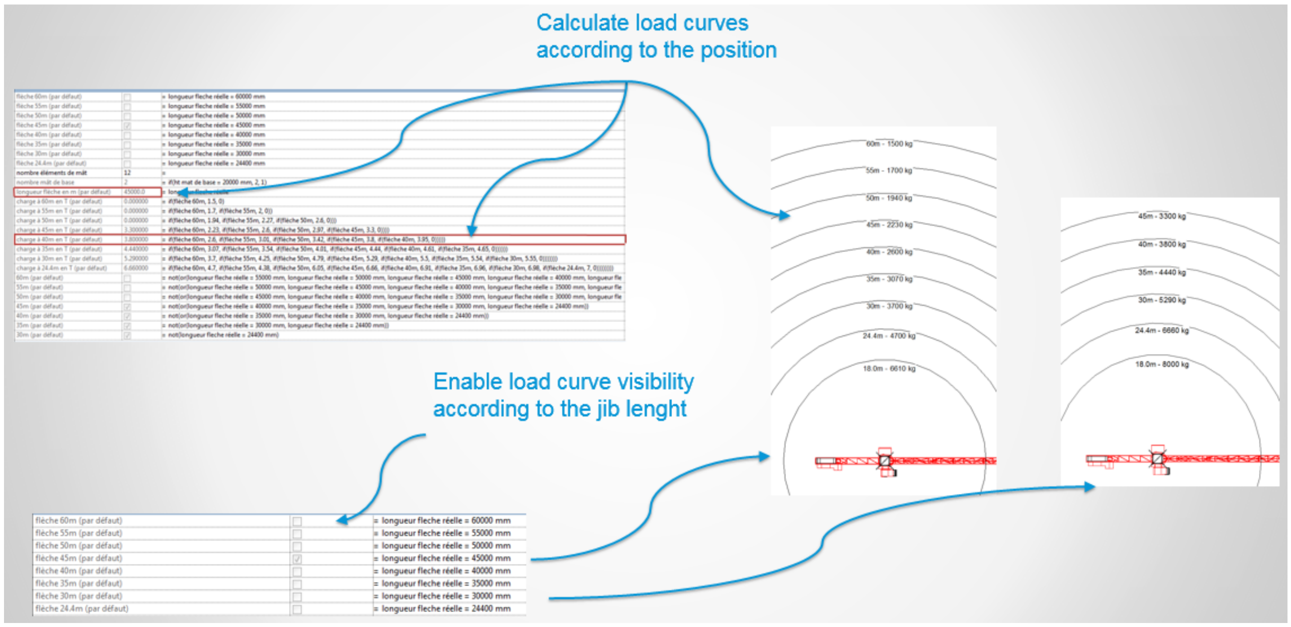




⊟ Modèles génériques	1	1	■ Noir	
Lignes cachées	1	1	■ Noir	Tiret
Saisissez le nouveau no...	1	1	■ Noir	Plein
VCF_Grue_Charges	1	1	■ Noir	Plein
VCF_Grue_ContreFleche	1	1	🟥 Rouge	Plein
VCF_Grue_Fleche1	1	1	□ Blanc	Plein
VCF_Grue_Fleche2	1	1	🟥 Rouge	Plein
VCF_Grue_Lest	1	1	■ Noir	Plein
VCF_Grue_Mat	1	1	🟦 Bleu	Plein
VCF_Grue_Pivot	1	1	■ Noir	Plein

3. Data management and maintenance

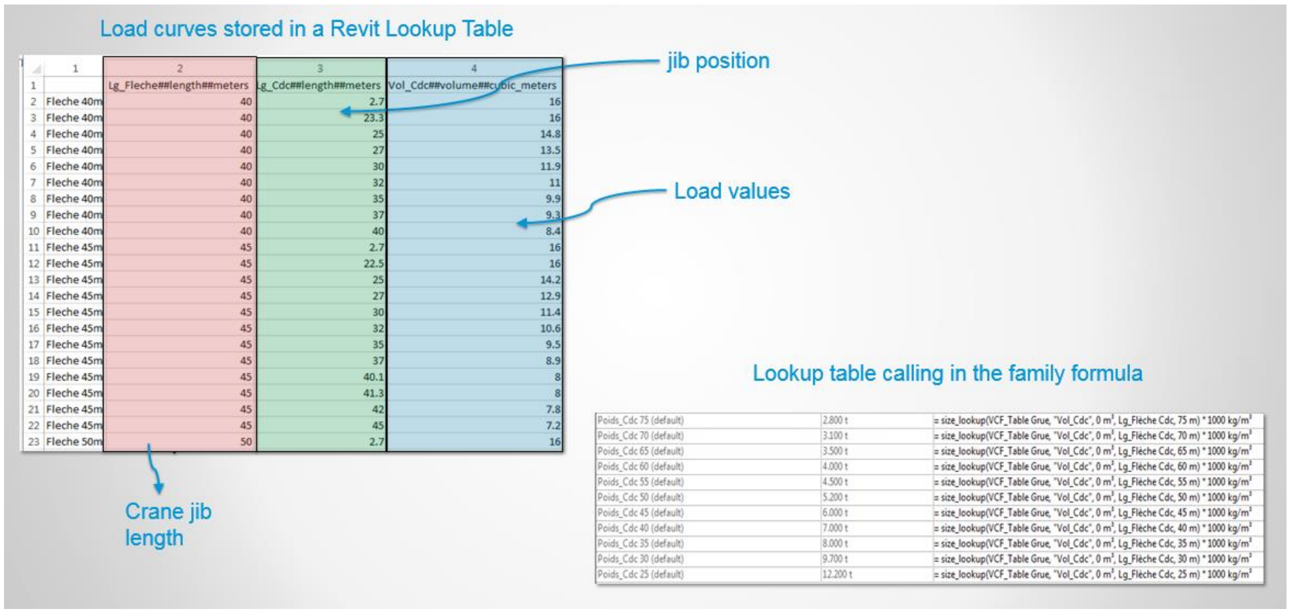
The tricky task in creating the tower crane family was to implement load curve information. First we tried to use conditional statement in the family parameter formula: it works but it is very tedious to implement and not really easy to maintain.



flèche 60m (par défaut)	☐	= longueur fleche réelle = 60000 mm
flèche 55m (par défaut)	☐	= longueur fleche réelle = 55000 mm
flèche 50m (par défaut)	☐	= longueur fleche réelle = 50000 mm
flèche 45m (par défaut)	☑	= longueur fleche réelle = 45000 mm
flèche 40m (par défaut)	☐	= longueur fleche réelle = 40000 mm
flèche 35m (par défaut)	☐	= longueur fleche réelle = 35000 mm
flèche 30m (par défaut)	☐	= longueur fleche réelle = 30000 mm
flèche 24.4m (par défaut)	☐	= longueur fleche réelle = 24400 mm
nombre éléments de mât	12	=
nombre mât de base	2	= if(ht mat de base = 20000 mm, 2, 1)
longueur flèche en m (par défaut)	45000.0	= lon[illegible] fleche réelle
charge à 60m en T (par défaut)	0.000000	= if(flèche 60m, 1.5, 0)
charge à 55m en T (par défaut)	0.000000	= if(flèche 60m, 1.7, if(flèche 55m, 2, 0))
charge à 50m en T (par défaut)	0.000000	= if(flèche 60m, 1.94, if(flèche 55m, 2.27, if(flèche 50m, 2.6, 0)))
charge à 45m en T (par défaut)	3.300000	= if(flèche 60m, 2.23, if(flèche 55m, 2.6, if(flèche 50m, 2.97, if(flèche 45m, 3.3, 0))))
charge à 40m en T (par défaut)	3.800000	= if(flèche 60m, 2.6, if(flèche 55m, 3.01, if(flèche 50m, 3.42, if(flèche 45m, 3.8, if(flèche 40m, 3.95, 0)))))
charge à 35m en T (par défaut)	4.440000	= if(flèche 60m, 3.07, if(flèche 55m, 3.54, if(flèche 50m, 4.01, if(flèche 45m, 4.44, if(flèche 40m, 4.61, if(flèche 35m, 4.65, 0))))))
charge à 30m en T (par défaut)	5.290000	= if(flèche 60m, 3.7, if(flèche 55m, 4.25, if(flèche 50m, 4.79, if(flèche 45m, 5.29, if(flèche 40m, 5.5, if(flèche 35m, 5.54, if(flèche 30m, 5.55, 0)))))))
charge à 24.4m en T (par défaut)	6.660000	= if(flèche 60m, 4.7, if(flèche 55m, 4.38, if(flèche 50m, 6.05, if(flèche 45m, 6.66, if(flèche 40m, 6.91, if(flèche 35m, 6.96, if(flèche 30m, 6.98, if(flèche 24.4m, 7, 0))))))))
60m (par défaut)	☐	= not(or(longueur fleche réelle = 55000 mm, longueur fleche réelle = 50000 mm, longueur fleche réelle = 45000 mm, longueur fleche réelle = 40000 mm, longueur fle
55m (par défaut)	☐	= not(or(longueur fleche réelle = 50000 mm, longueur fleche réelle = 45000 mm, longueur fleche réelle = 40000 mm, longueur fleche réelle = 35000 mm, longueur fle
50m (par défaut)	☐	= not(or(longueur fleche réelle = 45000 mm, longueur fleche réelle = 40000 mm, longueur fleche réelle = 35000 mm, longueur fleche réelle = 30000 mm, longueur fle
45m (par défaut)	☑	= not(or(longueur fleche réelle = 40000 mm, longueur fleche réelle = 35000 mm, longueur fleche réelle = 30000 mm, longueur fleche réelle = 24400 mm))
40m (par défaut)	☑	= not(or(longueur fleche réelle = 35000 mm, longueur fleche réelle = 30000 mm, longueur fleche réelle = 24400 mm))
35m (par défaut)	☑	= not(or(longueur fleche réelle = 30000 mm, longueur fleche réelle = 24400 mm))
30m (par défaut)	☑	= not(longueur fleche réelle = 24400 mm)

flèche 60m (par défaut)	☐	= longueur fleche réelle = 60000 mm
flèche 55m (par défaut)	☐	= longueur fleche réelle = 55000 mm
flèche 50m (par défaut)	☐	= longueur fleche réelle = 50000 mm
flèche 45m (par défaut)	☑	= longueur fleche réelle = 45000 mm
flèche 40m (par défaut)	☐	= longueur fleche réelle = 40000 mm
flèche 35m (par défaut)	☐	= longueur fleche réelle = 35000 mm
flèche 30m (par défaut)	☐	= longueur fleche réelle = 30000 mm
flèche 24.4m (par défaut)	☐	= longueur fleche réelle = 24400 mm

Then we decided to use the Lookup table feature available in the family editor: this tool enable us to store the load curves information in a more structured way in a .csv file. Then in the Revit parameter formula we can query specific information depending on parameter values (for example we can get the load according to the fork lift position and the crane jib length):

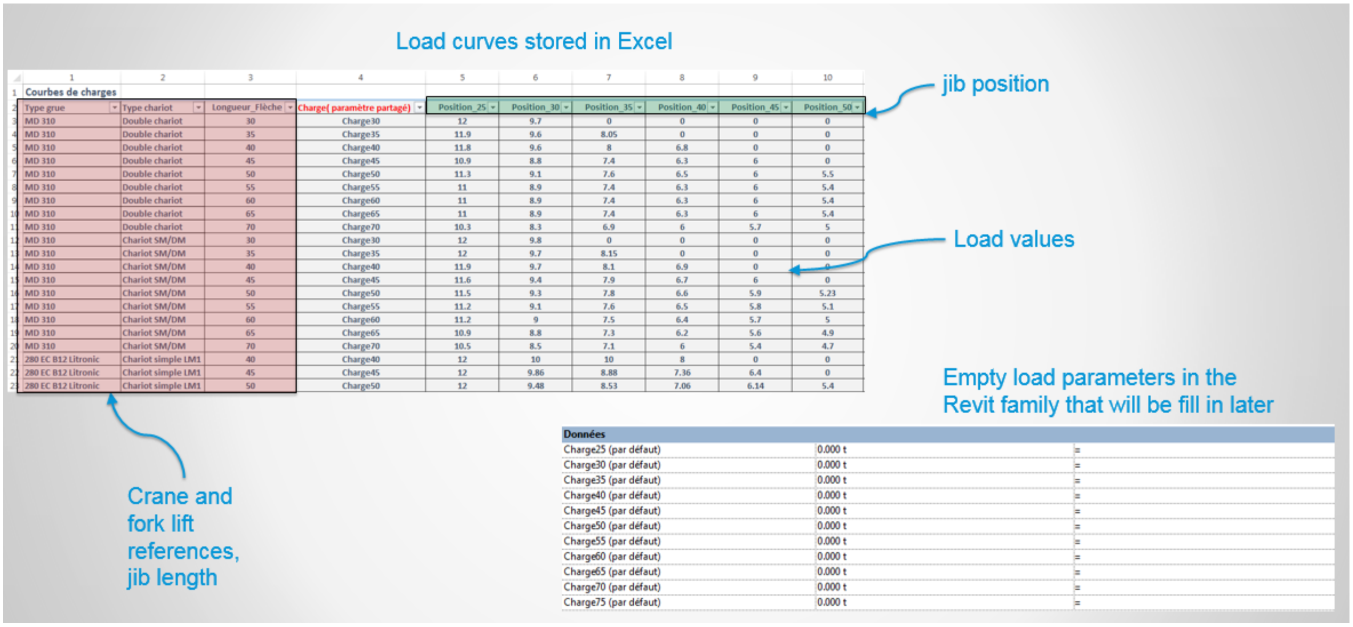

Load curves stored in a Revit Lookup Table
jib position
Load values
Crane jib length
Lookup table calling in the family formula

	1	2	3	4
1		Lg_Fleche##length##meters	Lg_Cdc##length##meters	Vol_Cdc##volume##cubic_meters
2	Fleche 40m	40	2.7	16
3	Fleche 40m	40	23.3	16
4	Fleche 40m	40	25	14.8
5	Fleche 40m	40	27	13.5
6	Fleche 40m	40	30	11.9
7	Fleche 40m	40	32	11
8	Fleche 40m	40	35	9.9
9	Fleche 40m	40	37	9.3
10	Fleche 40m	40	40	8.4
11	Fleche 45m	45	2.7	16
12	Fleche 45m	45	22.5	16
13	Fleche 45m	45	25	14.2
14	Fleche 45m	45	27	12.9
15	Fleche 45m	45	30	11.4
16	Fleche 45m	45	32	10.6
17	Fleche 45m	45	35	9.5
18	Fleche 45m	45	37	8.9
19	Fleche 45m	45	40.1	8
20	Fleche 45m	45	41.3	8
21	Fleche 45m	45	42	7.8
22	Fleche 45m	45	45	7.2
23	Fleche 50m	50	2.7	16

Poids_Cdc 75 (default)	2.800 t	= size_lookup(VCF_Table Grue, "Vol_Cdc", 0 m³, Lg_Flèche Cdc, 75 m) * 1000 kg/m³
Poids_Cdc 70 (default)	3.100 t	= size_lookup(VCF_Table Grue, "Vol_Cdc", 0 m³, Lg_Flèche Cdc, 70 m) * 1000 kg/m³
Poids_Cdc 65 (default)	3.500 t	= size_lookup(VCF_Table Grue, "Vol_Cdc", 0 m³, Lg_Flèche Cdc, 65 m) * 1000 kg/m³
Poids_Cdc 60 (default)	4.000 t	= size_lookup(VCF_Table Grue, "Vol_Cdc", 0 m³, Lg_Flèche Cdc, 60 m) * 1000 kg/m³
Poids_Cdc 55 (default)	4.500 t	= size_lookup(VCF_Table Grue, "Vol_Cdc", 0 m³, Lg_Flèche Cdc, 55 m) * 1000 kg/m³
Poids_Cdc 50 (default)	5.200 t	= size_lookup(VCF_Table Grue, "Vol_Cdc", 0 m³, Lg_Flèche Cdc, 50 m) * 1000 kg/m³
Poids_Cdc 45 (default)	6.000 t	= size_lookup(VCF_Table Grue, "Vol_Cdc", 0 m³, Lg_Flèche Cdc, 45 m) * 1000 kg/m³
Poids_Cdc 40 (default)	7.000 t	= size_lookup(VCF_Table Grue, "Vol_Cdc", 0 m³, Lg_Flèche Cdc, 40 m) * 1000 kg/m³
Poids_Cdc 35 (default)	8.000 t	= size_lookup(VCF_Table Grue, "Vol_Cdc", 0 m³, Lg_Flèche Cdc, 35 m) * 1000 kg/m³
Poids_Cdc 30 (default)	9.700 t	= size_lookup(VCF_Table Grue, "Vol_Cdc", 0 m³, Lg_Flèche Cdc, 30 m) * 1000 kg/m³
Poids_Cdc 25 (default)	12.200 t	= size_lookup(VCF_Table Grue, "Vol_Cdc", 0 m³, Lg_Flèche Cdc, 25 m) * 1000 kg/m³

This method works well but not very flexible: we can only deal with dimensional parameter (Text type parameter are not allowed for example) and we have to take care of the units. Furthermore, data are still embedded into the family, which means not sharable outside of Revit.

We finally tried the last method which provides much more flexibility and maintainability: all the data information are stored outside of the Revit family (Excel file but it can be a more sharable database like SQL Server) and a specific plugin has been created query the expected information and fill in the empty load parameters in Revit with the corresponding data:


Load curves stored in Excel
jib position
Load values
Empty load parameters in the Revit family that will be fill in later
Crane and fork lift references, jib length

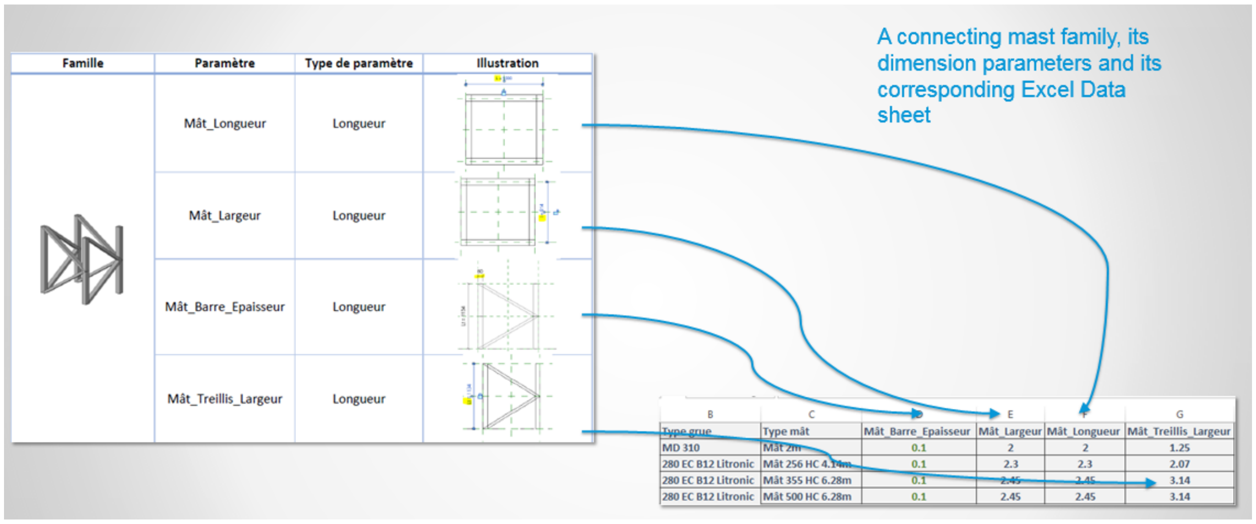
Courbes de charges

Type grue	Type chariot	Longueur_Flèche	Charge(paramètre partagé)	Position_25	Position_30	Position_35	Position_40	Position_45	Position_50
MD 310	Double chariot	30	Charge30	12	9.7	0	0	0	0
MD 310	Double chariot	35	Charge35	11.9	9.6	8.05	0	0	0
MD 310	Double chariot	40	Charge40	11.8	9.6	8	6.8	0	0
MD 310	Double chariot	45	Charge45	10.9	8.8	7.4	6.3	6	0
MD 310	Double chariot	50	Charge50	11.3	9.1	7.6	6.5	6	5.5
MD 310	Double chariot	55	Charge55	11	8.9	7.4	6.3	6	5.4
MD 310	Double chariot	60	Charge60	11	8.9	7.4	6.3	6	5.4
MD 310	Double chariot	65	Charge65	11	8.9	7.4	6.3	6	5.4
MD 310	Double chariot	70	Charge70	10.3	8.3	6.9	6	5.7	5
MD 310	Chariot SM/DM	30	Charge30	12	9.8	0	0	0	0
MD 310	Chariot SM/DM	35	Charge35	12	9.7	8.15	0	0	0
MD 310	Chariot SM/DM	40	Charge40	11.9	9.7	8.1	6.9	0	0
MD 310	Chariot SM/DM	45	Charge45	11.6	9.4	7.9	6.7	6	0
MD 310	Chariot SM/DM	50	Charge50	11.5	9.3	7.8	6.6	5.9	5.23
MD 310	Chariot SM/DM	55	Charge55	11.2	9.1	7.6	6.5	5.8	5.1
MD 310	Chariot SM/DM	60	Charge60	11.2	9	7.5	6.4	5.7	5
MD 310	Chariot SM/DM	65	Charge65	10.9	8.8	7.3	6.2	5.6	4.9
MD 310	Chariot SM/DM	70	Charge70	10.5	8.5	7.1	6	5.4	4.7
280 EC B12 Litronic	Chariot simple LM1	40	Charge40	12	10	10	8	0	0
280 EC B12 Litronic	Chariot simple LM1	45	Charge45	12	9.86	8.88	7.36	6.4	0
280 EC B12 Litronic	Chariot simple LM1	50	Charge50	12	9.48	8.53	7.06	6.14	5.4

Données		
Charge25 (par défaut)	0.000 t	=
Charge30 (par défaut)	0.000 t	=
Charge35 (par défaut)	0.000 t	=
Charge40 (par défaut)	0.000 t	=
Charge45 (par défaut)	0.000 t	=
Charge50 (par défaut)	0.000 t	=
Charge55 (par défaut)	0.000 t	=
Charge60 (par défaut)	0.000 t	=
Charge65 (par défaut)	0.000 t	=
Charge70 (par défaut)	0.000 t	=
Charge75 (par défaut)	0.000 t	=

4. Family maintenance by creating generic components

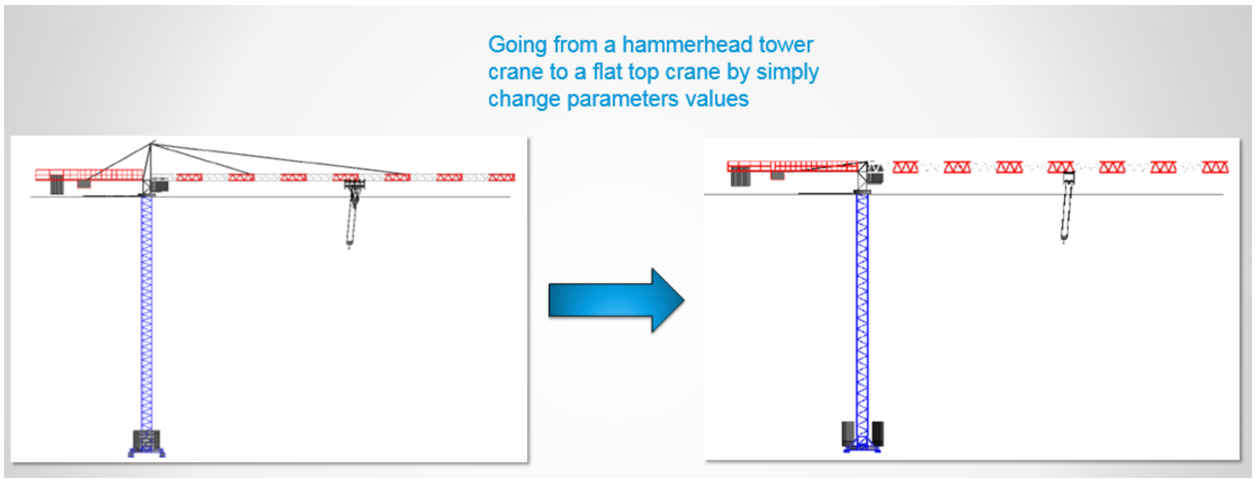
We also decided to make the tower crane highly parametric in order to maintain less families and to handle all standards and specific cases in only one family component. Therefore each piece of the crane has a bunch of dimension parameters and/or visibility settings:



A connecting mast family, its dimension parameters and its corresponding Excel Data sheet

Famille	Paramètre	Type de paramètre	Illustration
	Mât_Longueur	Longueur	
	Mât_Largeur	Longueur	
	Mât_Barre_Epaisseur	Longueur	
	Mât_Treillis_Largeur	Longueur	

B	C	D	E	F	G
Type grue	Type mât	Mât_Barre_Epaisseur	Mât_Largeur	Mât_Longueur	Mât_Treillis_Largeur
MD 310	Mât 2m	0.1	2	2	1.25
280 EC B12 Litronic	Mât 256 HC 4.14m	0.1	2.3	2.3	2.07
280 EC B12 Litronic	Mât 355 HC 6.28m	0.1	2.45	2.45	3.14
280 EC B12 Litronic	Mât 500 HC 6.28m	0.1	2.45	2.45	3.14



Going from a hammerhead tower crane to a flat top crane by simply change parameters values

5.2 Safety platform

The global requirements specification for this equipment library are:

1. **Include all the features which enable our engineers to realize safety layout drawing**
   These features are platform foots/extension rules, minimum space requirement from the façade, etc.

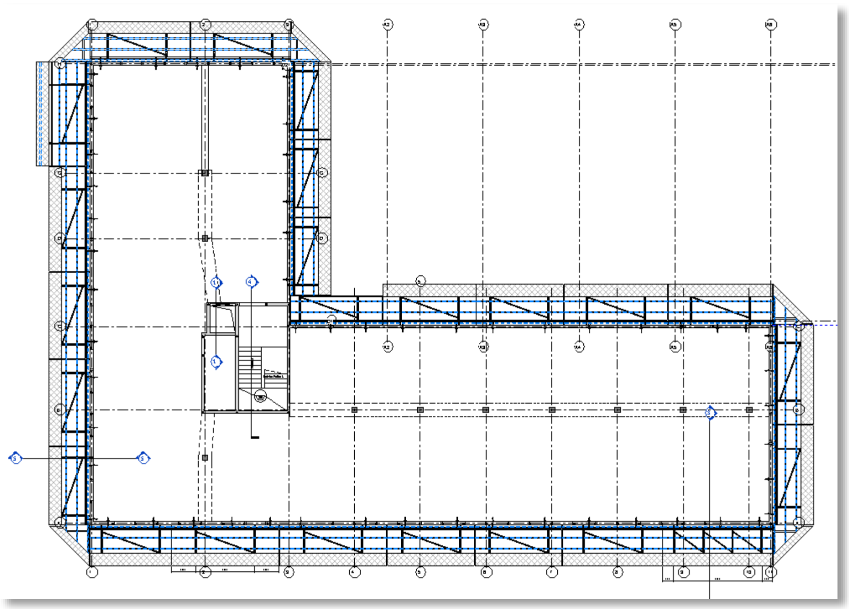


FIGURE 50 SAFETY PLATFORM LAYOUT PLAN

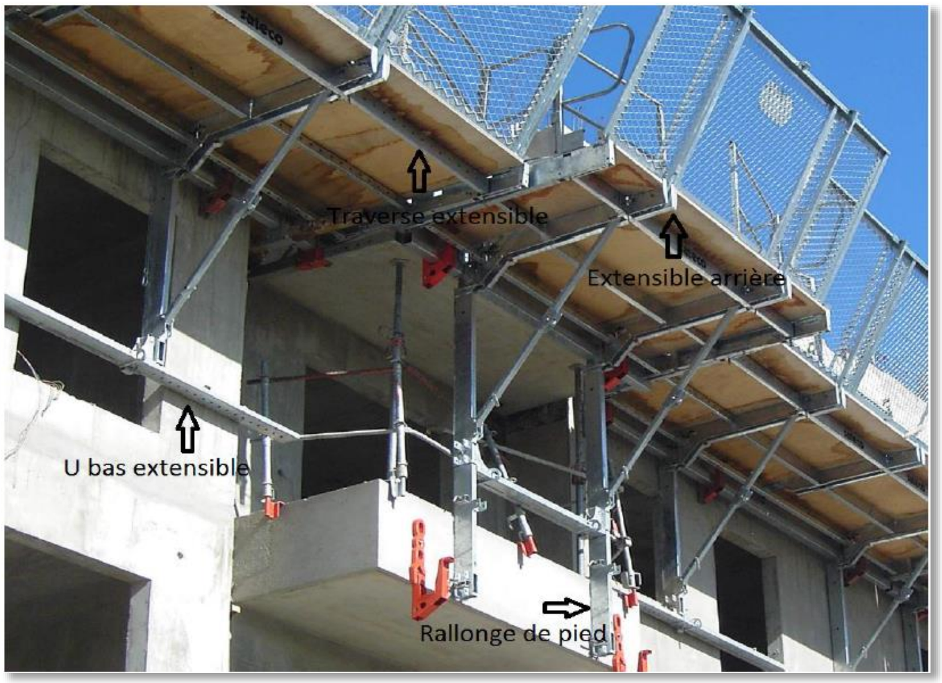
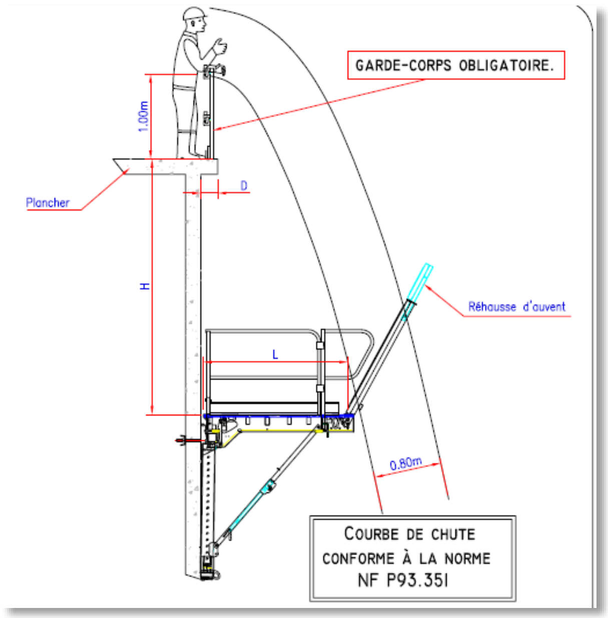
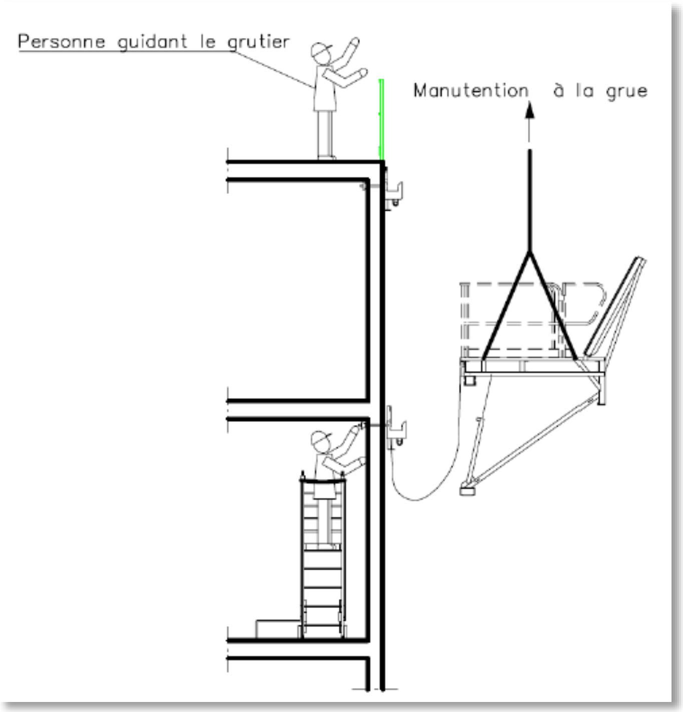


FIGURE 51 SAFETY PLATFORM LAYOUT FEATURES

2. **Include building site safety and circulation rules**
   Like falling curves, safety railings, rear extension:

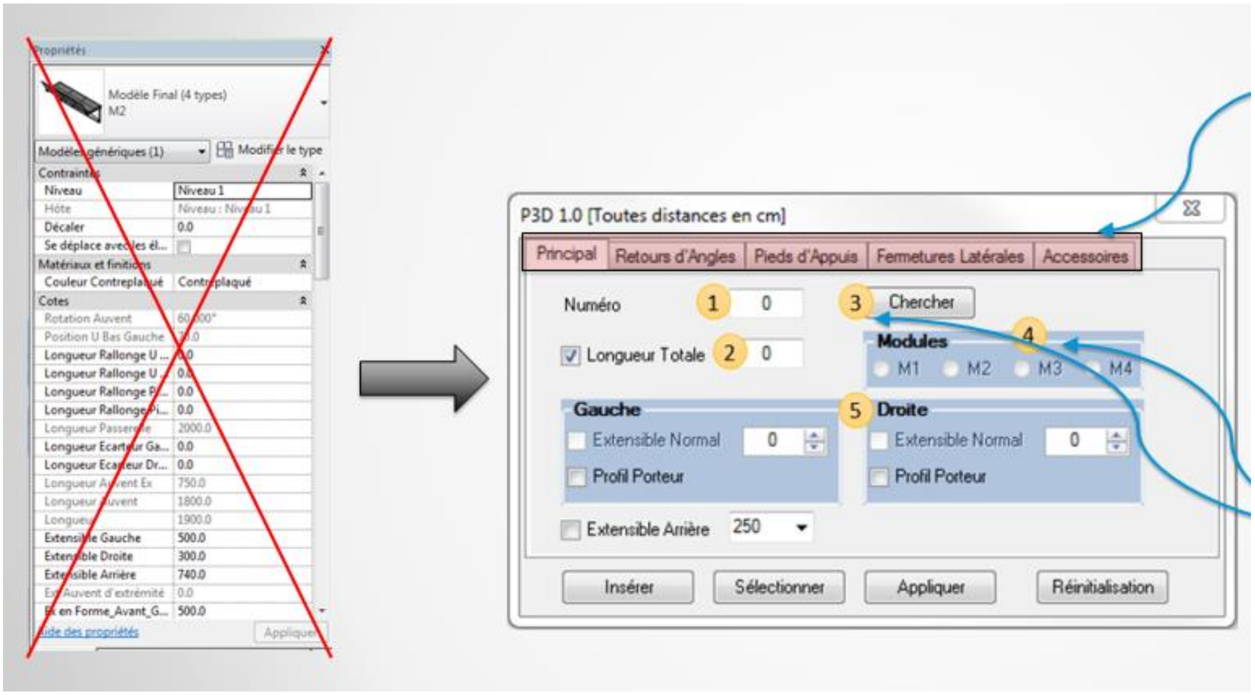


3. **Include lifting rings position**
   In order to check if the safety platform can be released



4. **Create a friendly user interface**

Because this equipment is highly parametric, we took the decision to develop a user interface with Revit API.



Propriétés

Modèle Final (4 types)
M2

Modèles génériques (1) — Modifier le type

Contraintes	
Niveau	Niveau 1
Hôte	Niveau : Niveau 1
Décaler	0.0
Se déplace avec les él...	☐
Matériaux et finitions	
Couleur Contreplaqué	Contreplaqué
Cotes	
Rotation Auvent	60.000°
Position U Bas Gauche	70.0
Longueur Rallonge U ...	0.0
Longueur Rallonge U ...	0.0
Longueur Rallonge P...	0.0
Longueur Rallonge Pi...	0.0
Longueur Passerelle	2000.0
Longueur Ecarteur Ga...	0.0
Longueur Ecarteur Dr...	0.0
Longueur Auvent Ex	750.0
Longueur Auvent	1800.0
Longueur	1900.0
Extensible Gauche	500.0
Extensible Droite	300.0
Extensible Arrière	740.0
Ex Auvent d'extrémité	0.0
Ex en Forme_Avant_G...	500.0

Aide des propriétés — Appliquer

P3D 1.0 [Toutes distances en cm]

Principal | Retours d'Angles | Pieds d'Appuis | Fermetures Latérales | Accessoires

Numéro 1 0 — 3 Chercher

☑ Longueur Totale 2 0 — 4 Modules: M1 M2 M3 M4

Gauche: ☐ Extensible Normal 0; ☐ Profil Porteur — 5 Droite: ☐ Extensible Normal 0; ☐ Profil Porteur

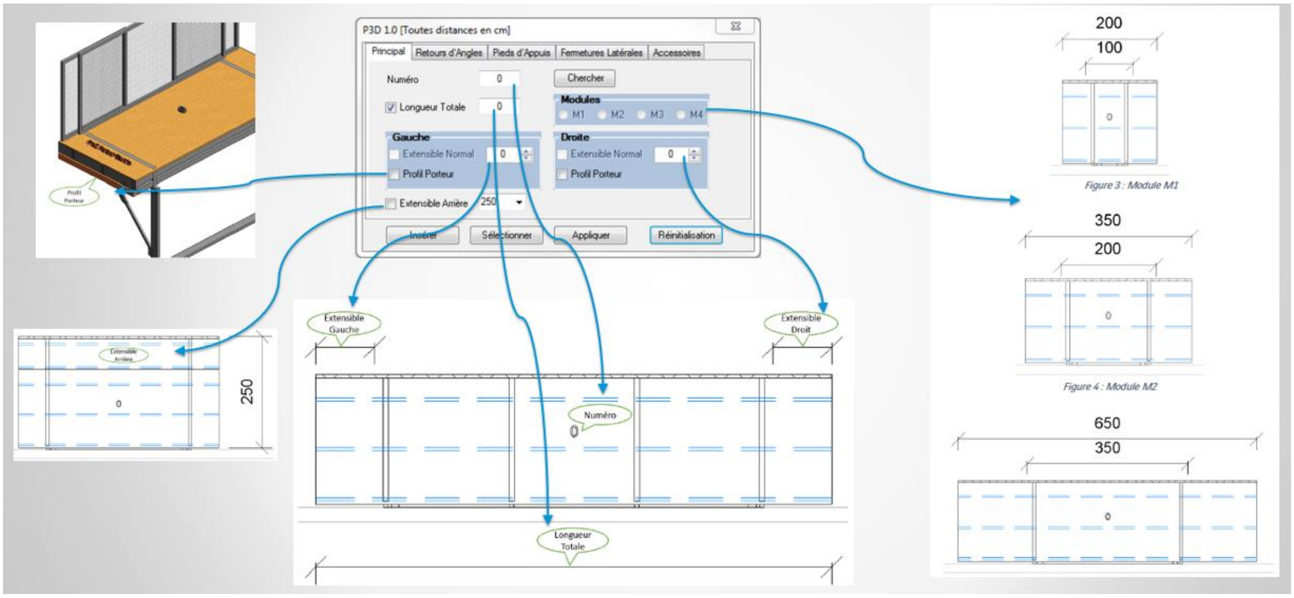
☐ Extensible Arrière 250

Insérer | Sélectionner | Appliquer | Réinitialisation

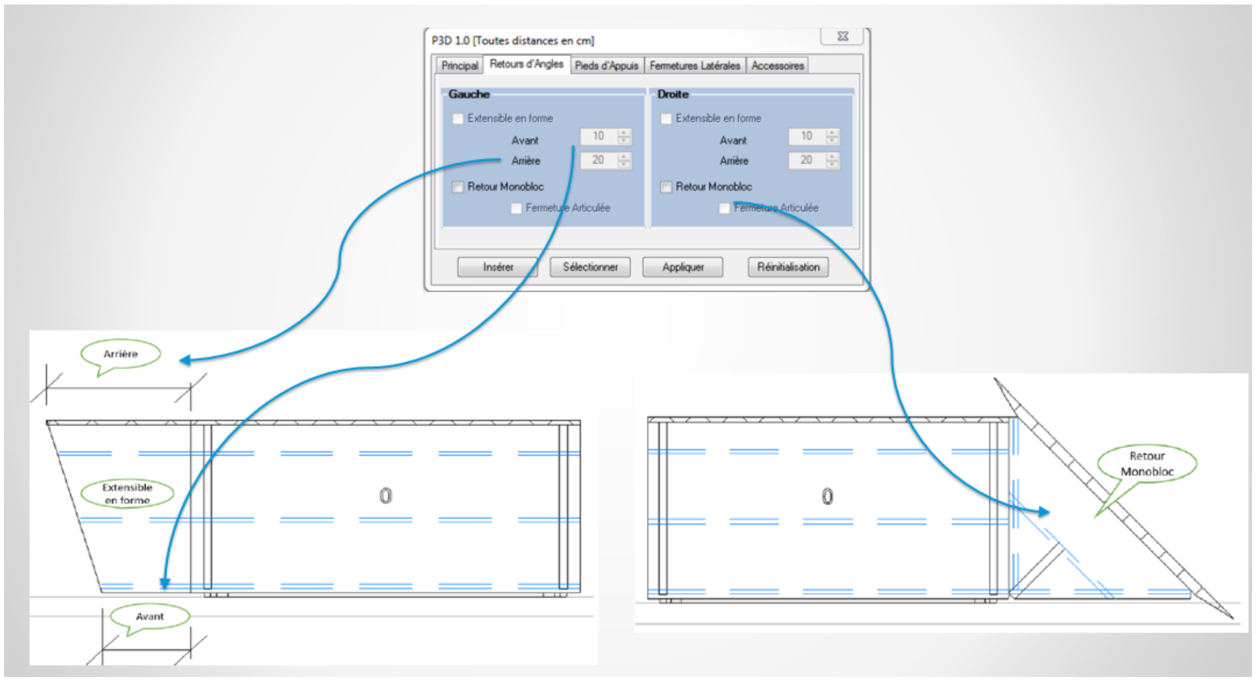
FIGURE 52 PANEL PROPERTIES REPLACEMENT BY A FRIENDLIER UI

A tab organization has been made in order to minimize the window application. A thematic classification has been done:

- **Tab 1:** General dimensions (global length, modules, end and rear extensions):



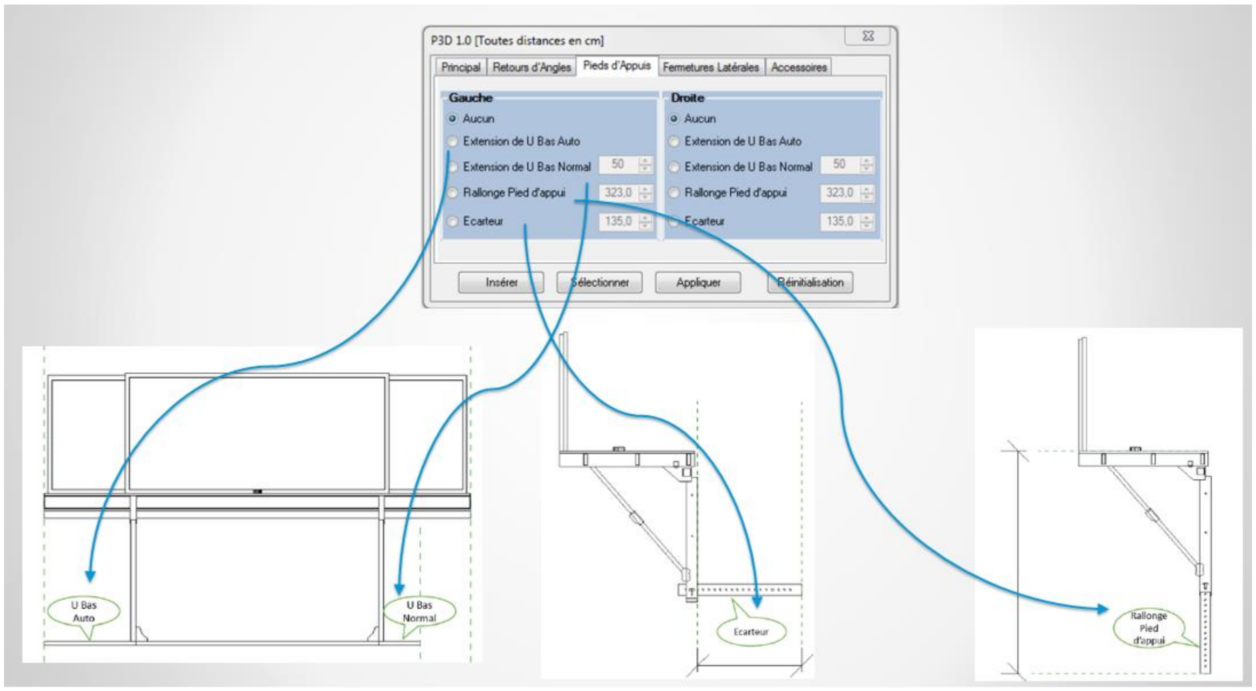
- **Tab 2:** Angular-end extensions and angled side:



Gauche
☐ Extensible en forme
Avant 10
Arrière 20
☐ Retour Monobloc
☐ Fermeture Articulée

Droite
☐ Extensible en forme
Avant 10
Arrière 20
☐ Retour Monobloc
☐ Fermeture Articulée

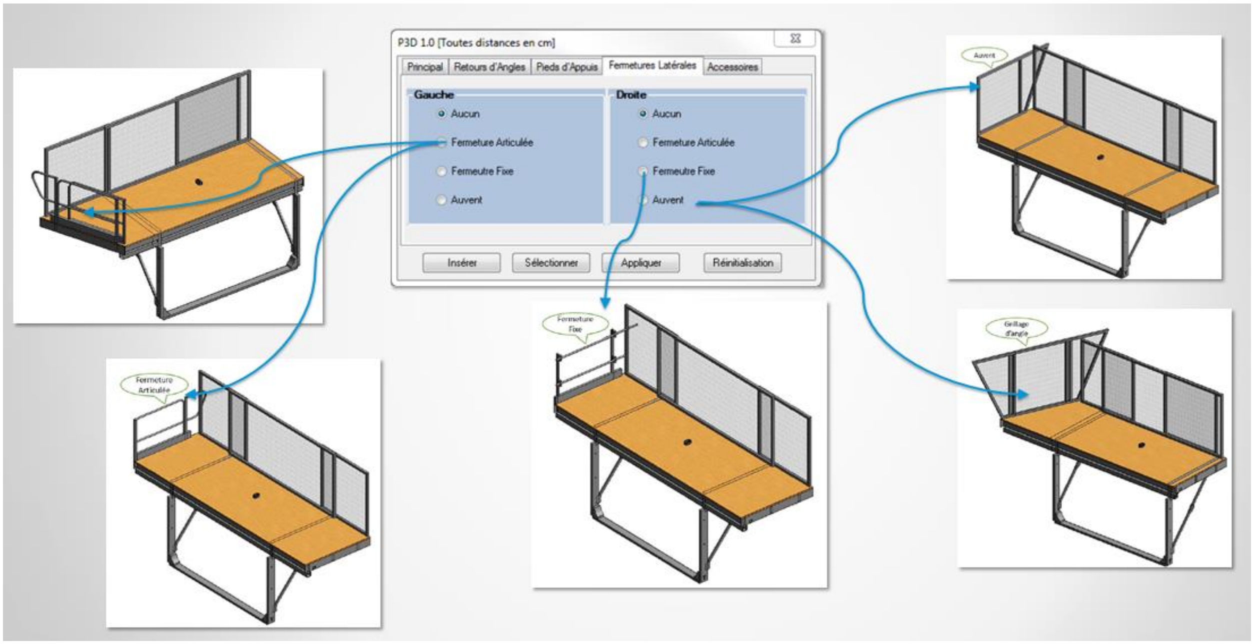
- **Tab 3:** Platform foot (foot extension, lower spacer, and extensible lower U):



Gauche
[● Aucun]
○ Extension de U Bas Auto
○ Extension de U Bas Normal 50
○ Rallonge Pied d'appui 323,0
○ Ecarteur 135,0

Droite
[● Aucun]
○ Extension de U Bas Auto
○ Extension de U Bas Normal 50
○ Rallonge Pied d'appui 323,0
○ Ecarteur 135,0

- **Tab 4:** Locking system at platform's ends (fixed or removable):



P3D 1.0 [Toutes distances en cm]

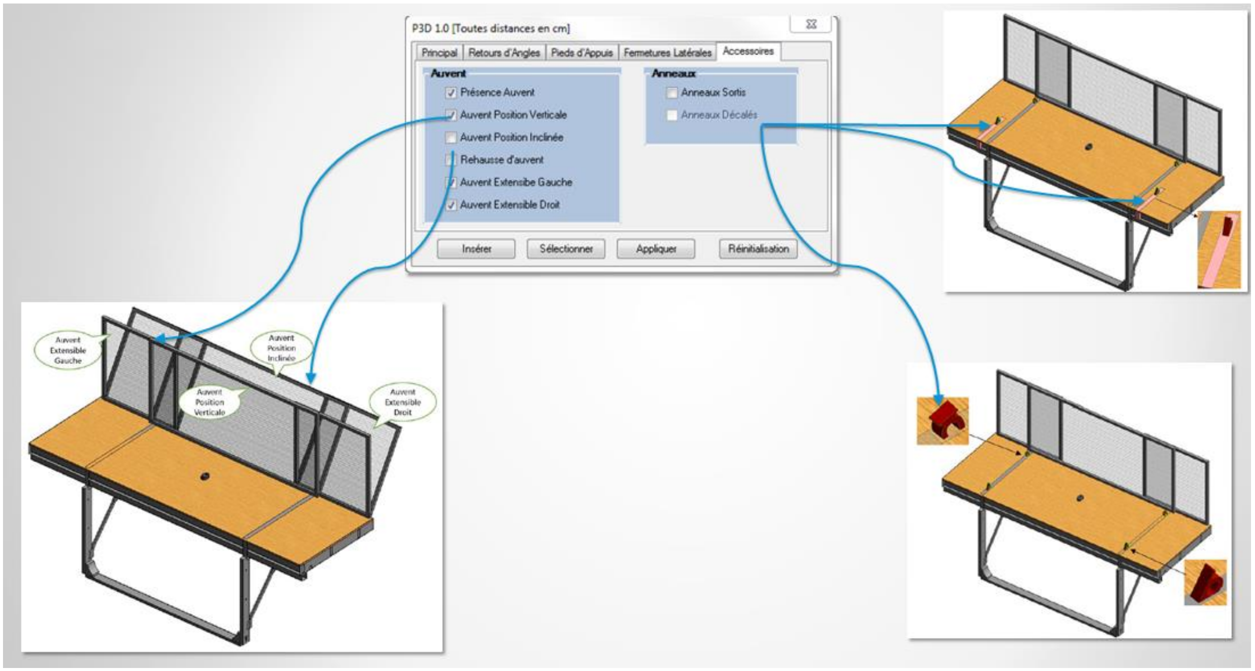
Principal | Retours d'Angles | Pieds d'Appuis | Fermetures Latérales | Accessoires

Gauche
- [✓ Aucun]
- Fermeture Articulée
- Fermeutre Fixe
- Auvent

Droite
- [✓ Aucun]
- Fermeture Articulée
- Fermeutre Fixe
- Auvent

Insérer | Sélectionner | Appliquer | Réinitialisation

- **Tab 5:** Other options like lifting rings position, vertical or sloping grid:



P3D 1.0 [Toutes distances en cm]

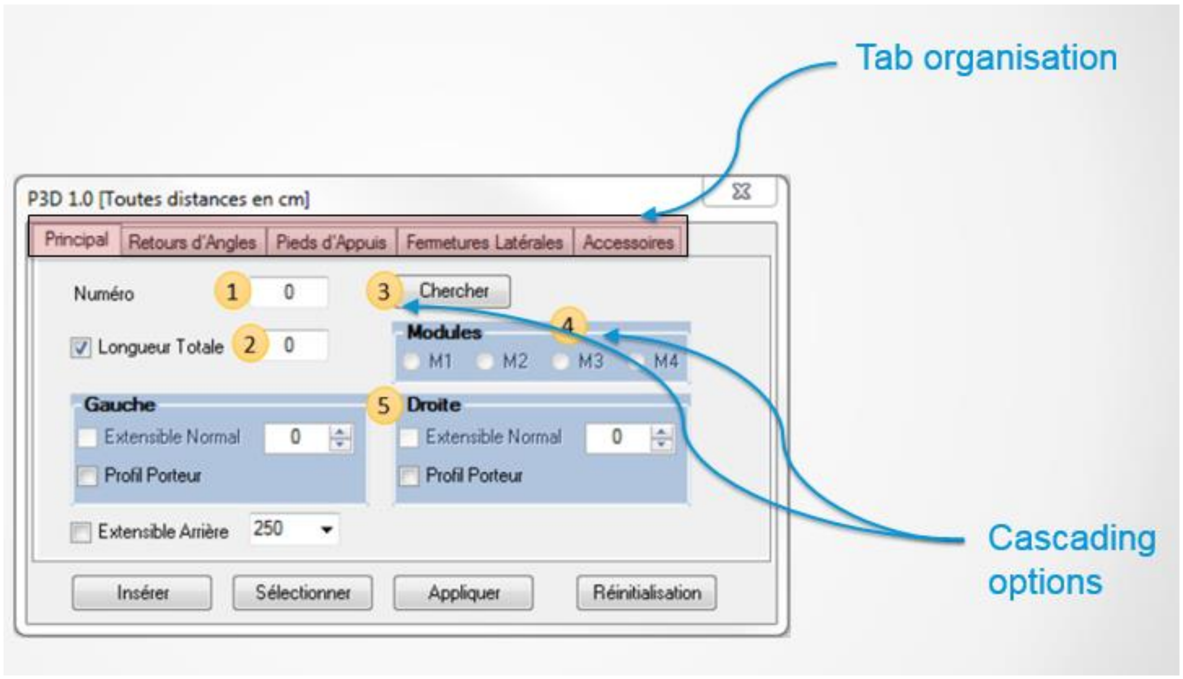
Principal | Retours d'Angles | Pieds d'Appuis | Fermetures Latérales | Accessoires

Auvent
- [✓] Présence Auvent
- [✓] Auvent Position Verticale
- [ ] Auvent Position Inclinée
- [ ] Rehausse d'auvent
- [✓] Auvent Extensibe Gauche
- [✓] Auvent Extensible Droit

Anneaux
- [ ] Anneaux Sortis
- [ ] Anneaux Décalés

Insérer | Sélectionner | Appliquer | Réinitialisation

Through logical path and cascading choices, the UI helps engineers make the best and safety solution. For example the user can enter the total platform length and as a result the UI will suggest the available modules. The user can also increment the left or right extension, and the opposite extension will decrement in order to respect the total length:



P3D 1.0 [Toutes distances en cm]

Principal | Retours d'Angles | Pieds d'Appuis | Fermetures Latérales | Accessoires

Numéro 0
Chercher
☑ Longueur Totale 0
Modules: M1, M2, M3, M4
Gauche: ☐ Extensible Normal 0; ☐ Profil Porteur
Droite: ☐ Extensible Normal 0; ☐ Profil Porteur
☐ Extensible Arrière 250

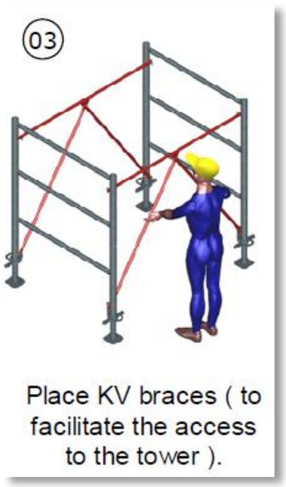
Insérer | Sélectionner | Appliquer | Réinitialisation

FIGURE 53 CASCADING CHOICES WHICH GUIDE USER STEP BY STEP

5.3 Shoring tower

The global requirements specification for this equipment library are:

1. **Tower accessibility:**
   The level of details in 3D and 2D must be enough in order to indicate the tower entry – it is often materialized by a special frame – in order to check circulations depending on its position and environment:

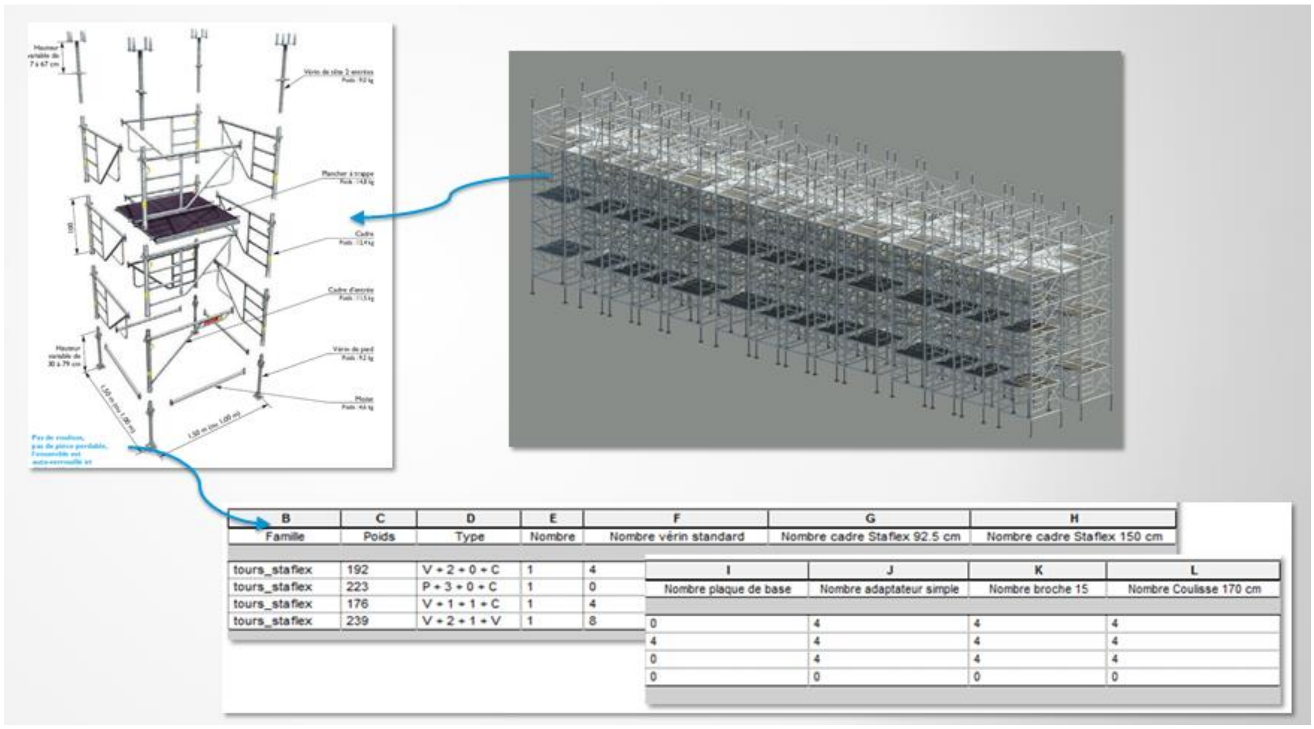


2. **Setup and takedown time:**
   This information is stored into Revit parameters and vary depending on the tower height (Formula use). This information is relevant for planning purpose

3. **Presence of safety accessories**
   Safety railings, trapdoor platform position, and presence of bracings: this is important to be compliant with the safety and accessibility standards and to be sure that nothing is missing during the delivery:

> 12- Terminer la mise en place et le réglage des 2 vérins de tête.

4. **Parts scheduling**
   Because shoring tower are composed of many parts, it is sometimes difficult to deliver an accurate scheduling based on a shoring layout plan. With Revit, it is possible to get this quantity either with shared nested family or shared parameter conditional values:



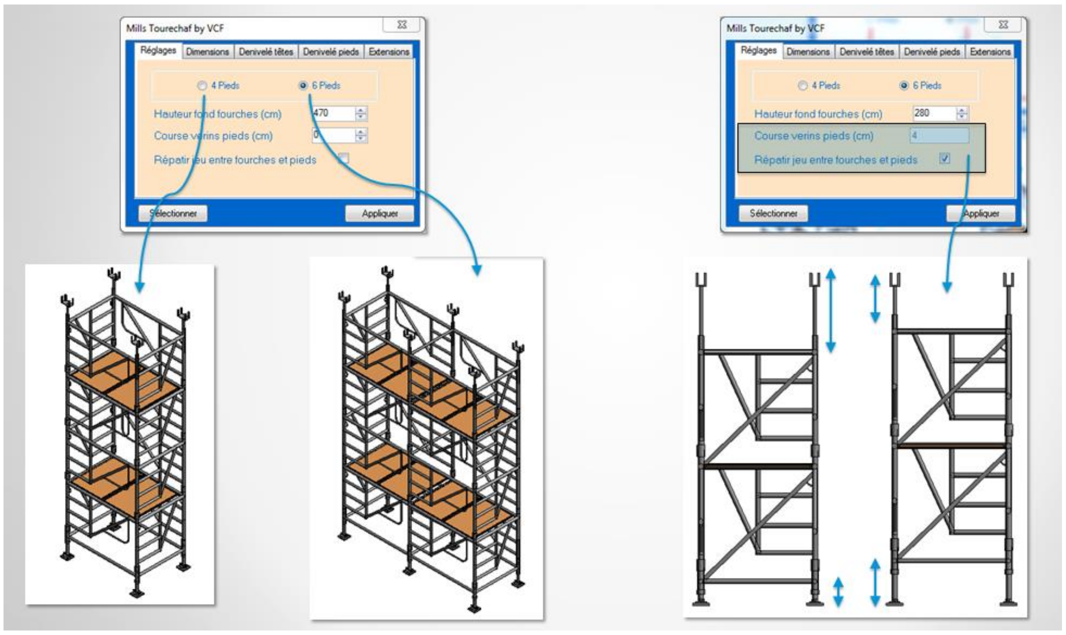
B	C	D	E	F	G	H
Famille	Poids	Type	Nombre	Nombre vérin standard	Nombre cadre Staflex 92.5 cm	Nombre cadre Staflex 150 cm
tours_staflex	192	V + 2 + 0 + C	1	4		
tours_staflex	223	P + 3 + 0 + C	1	0		
tours_staflex	176	V + 1 + 1 + C	1	4		
tours_staflex	239	V + 2 + 1 + V	1	8		

I	J	K	L
Nombre plaque de base	Nombre adaptateur simple	Nombre broche 15	Nombre Coulisse 170 cm
0	4	4	4
4	4	4	4
0	4	4	4
0	0	0	0

5. User interface

We also choose a tab organization within the user interface in order to have a thematic user input classification:

- **Tab 1:** used to choose the number of legs, the height of fork bottom, and the distribution of distance between base jack and fork:



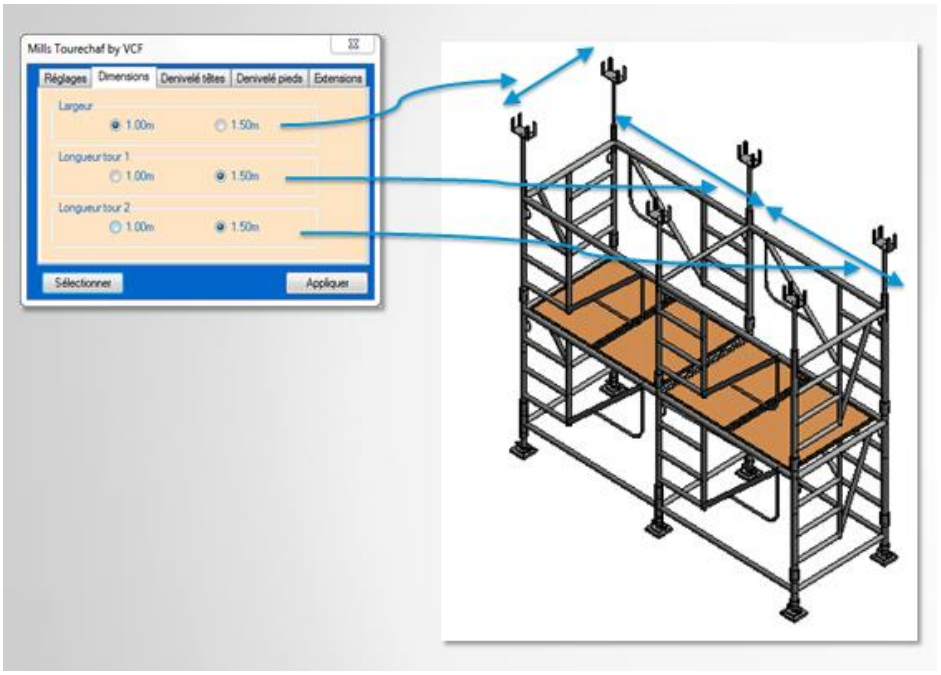
Mills Tourechaf by VCF

○ 4 Pieds [✓ ● 6 Pieds]
Hauteur fond fourches (cm): 470
Course verins pieds (cm): 0
Répatir jeu entre fourches et pieds: ☐

Mills Tourechaf by VCF

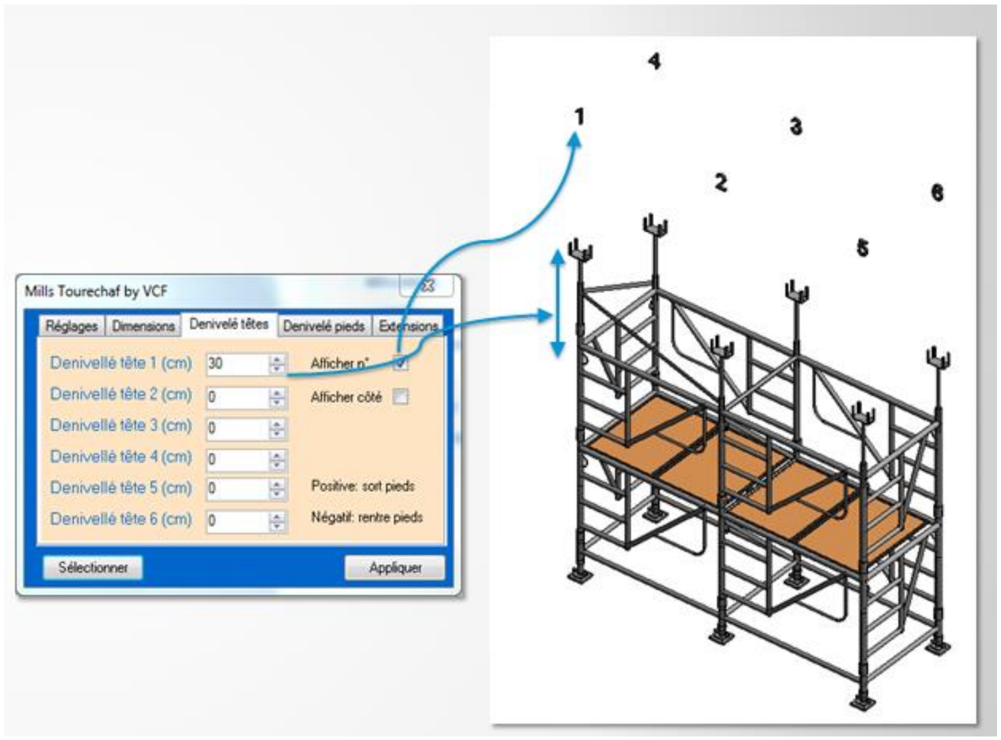
○ 4 Pieds [✓ ● 6 Pieds]
Hauteur fond fourches (cm): 280
Course verins pieds (cm): 4
Répatir jeu entre fourches et pieds: ☑

- **Tab 2:** used to apply the general footprint dimensions:



Largeur: [✓ ● 1.00m] ○ 1.50m
Longueur tour 1: ○ 1.00m [✓ ● 1.50m]
Longueur tour 2: ○ 1.00m [✓ ● 1.50m]

- **Tab 3:** used to define the individual fork extension or retraction. A number identify on which fork we are working on. If the extension is greater than a specific distance, a brace is automatically placed:

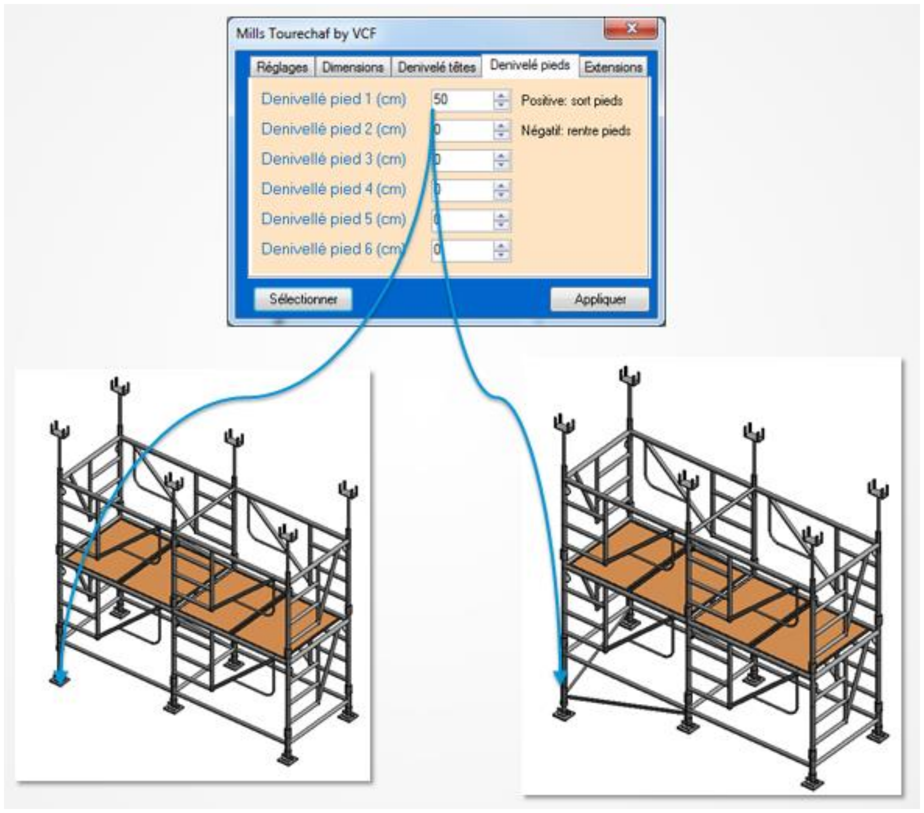

Mills Tourechaf by VCF

Denivellé tête 1 (cm) 30
Denivellé tête 2 (cm) 0
Denivellé tête 3 (cm) 0
Denivellé tête 4 (cm) 0
Denivellé tête 5 (cm) 0
Denivellé tête 6 (cm) 0

Afficher n° ✓
Afficher côté

Positive: sort pieds
Négatif: rentre pieds

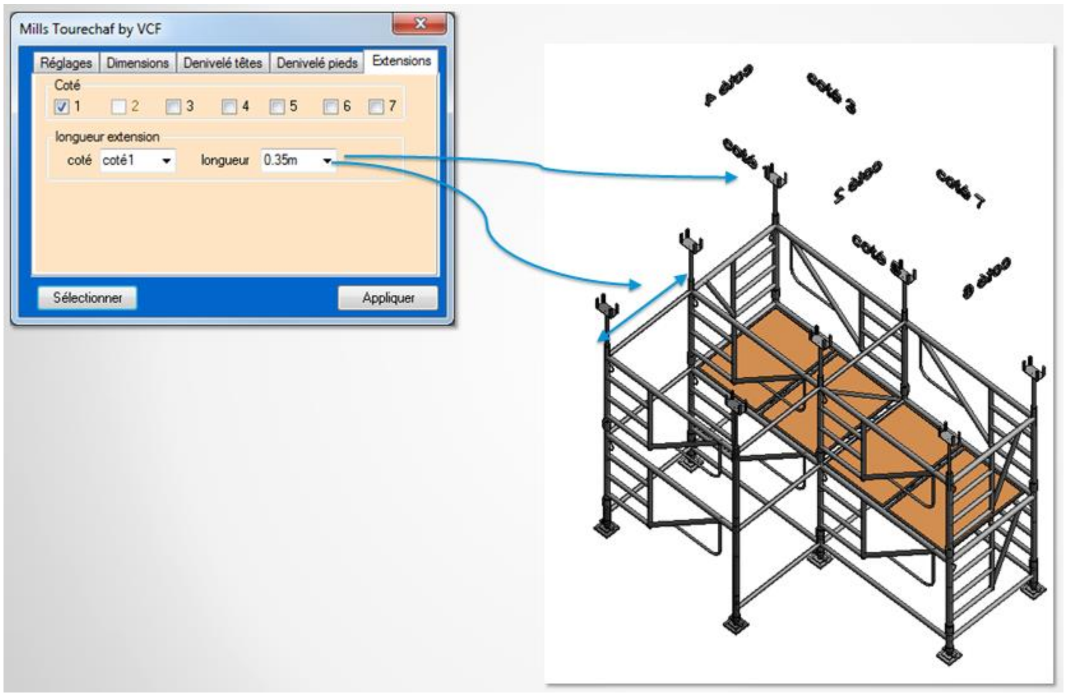
- **Tab 4:** Same logic is applied for the base jack:


Mills Tourechaf by VCF

Denivellé pied 1 (cm) 50
Denivellé pied 2 (cm) 0
Denivellé pied 3 (cm) 0
Denivellé pied 4 (cm) 0
Denivellé pied 5 (cm) 0
Denivellé pied 6 (cm) 0

Positive: sort pieds
Négatif: rentre pieds

- **Tab 5:** we can add extensions by picking the side and choose one of the available length in the combo box:



Coté: [✓ 1] 2 3 4 5 6 7

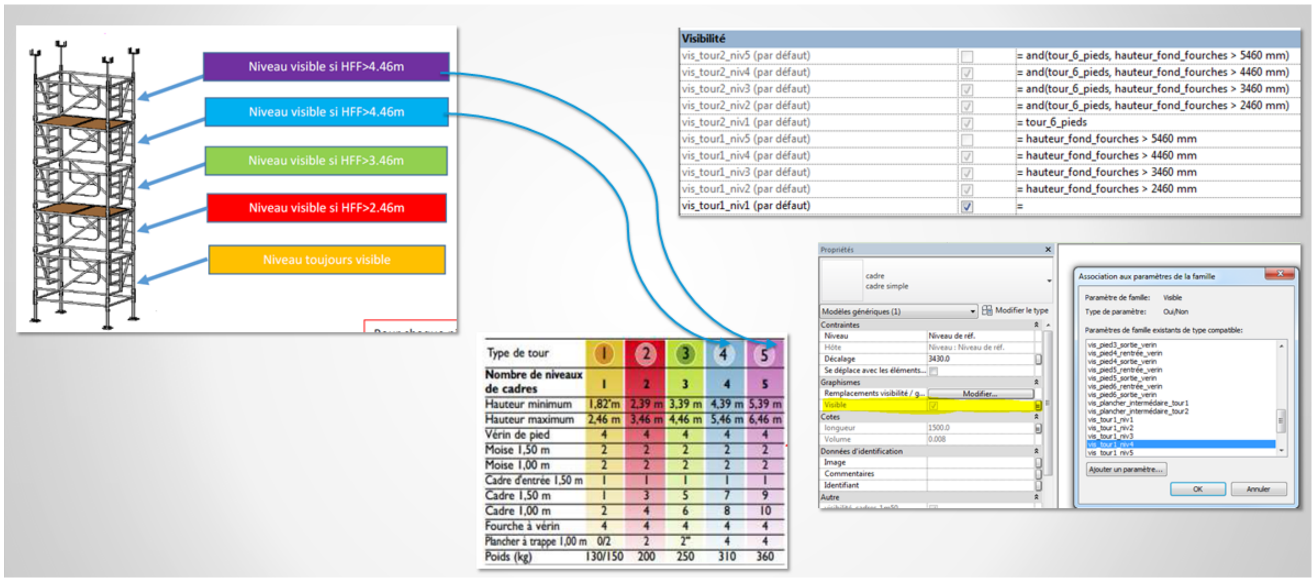
longueur extension

coté: coté1 longueur: 0.35m

6. Deal with frames superposition

The tricky part on creating the shoring tower family was to show/hide tower frames depending on the height of fork bottom. Two ways have been tested:

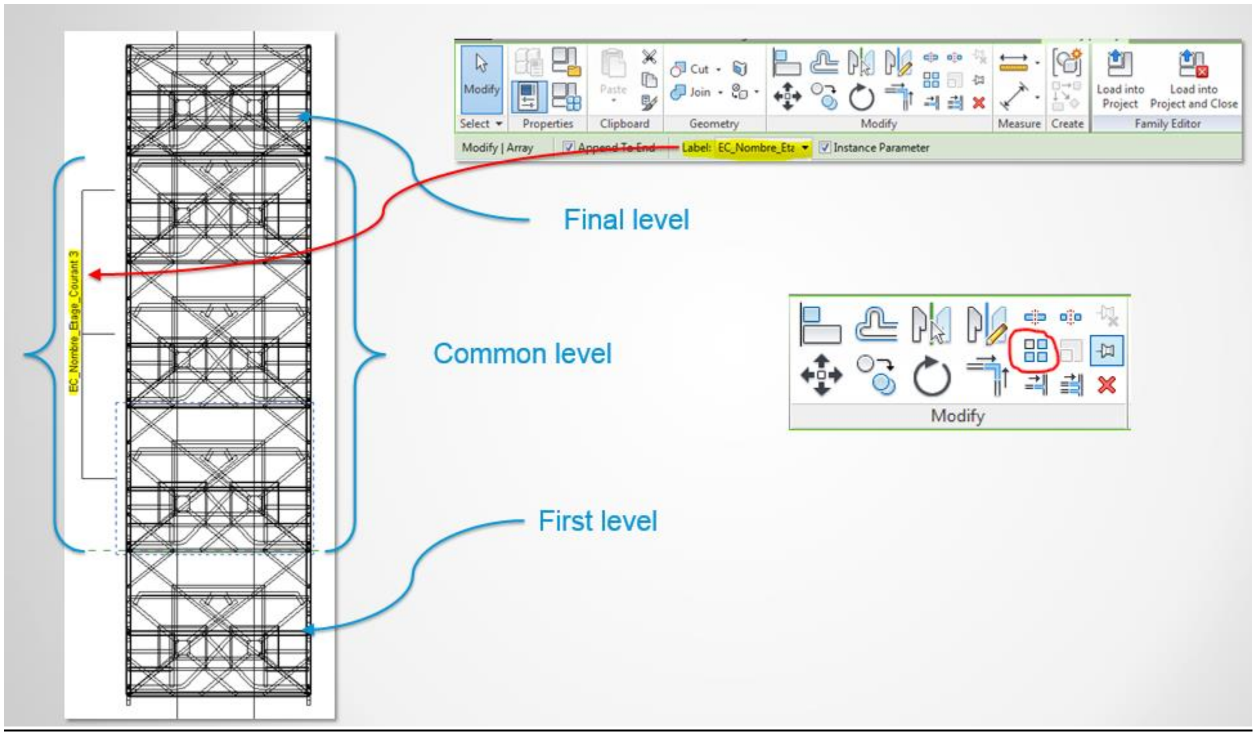
- <u>First method: Relate family visibility parameter to formula</u>: for each level, the frame's visibility property value is set to a formula. This formula specify if the visibility parameter is checked or not depending on the height of fork bottom


Niveau visible si HFF>4.46m
Niveau visible si HFF>4.46m
Niveau visible si HFF>3.46m
Niveau visible si HFF>2.46m
Niveau toujours visible
Propriétés
cadre
cadre simple
Modèles génériques (1)
Modifier le type
Association aux paramètres de la famille
Paramètre de famille: Visible
Type de paramètre: Oui/Non
OK
Annuler

Visibilité		
vis_tour2_niv5 (par défaut)	☐	= and(tour_6_pieds, hauteur_fond_fourches > 5460 mm)
vis_tour2_niv4 (par défaut)	✓	= and(tour_6_pieds, hauteur_fond_fourches > 4460 mm)
vis_tour2_niv3 (par défaut)	✓	= and(tour_6_pieds, hauteur_fond_fourches > 3460 mm)
vis_tour2_niv2 (par défaut)	✓	= and(tour_6_pieds, hauteur_fond_fourches > 2460 mm)
vis_tour2_niv1 (par défaut)	✓	= tour_6_pieds
vis_tour1_niv5 (par défaut)	☐	= hauteur_fond_fourches > 5460 mm
vis_tour1_niv4 (par défaut)	✓	= hauteur_fond_fourches > 4460 mm
vis_tour1_niv3 (par défaut)	✓	= hauteur_fond_fourches > 3460 mm
vis_tour1_niv2 (par défaut)	✓	= hauteur_fond_fourches > 2460 mm
vis_tour1_niv1 (par défaut)	✓	=

Type de tour	1	2	3	4	5
Nombre de niveaux de cadres	**1**	**2**	**3**	**4**	**5**
Hauteur minimum	1,82'm	2,39 m	3,39 m	4,39 m	5,39 m
Hauteur maximum	2,46 m	3,46 m	4,46 m	5,46 m	6,46 m
Vérin de pied	4	4	4	4	4
Moise 1,50 m	2	2	2	2	2
Moise 1,00 m	2	2	2	2	2
Cadre d'entrée 1,50 m	1	1	1	1	1
Cadre 1,50 m	1	3	5	7	9
Cadre 1,00 m	2	4	6	8	10
Fourche à vérin	4	4	4	4	4
Plancher à trappe 1,00 m	0/2	2	2"	4	4
Poids (kg)	130/150	200	250	310	360

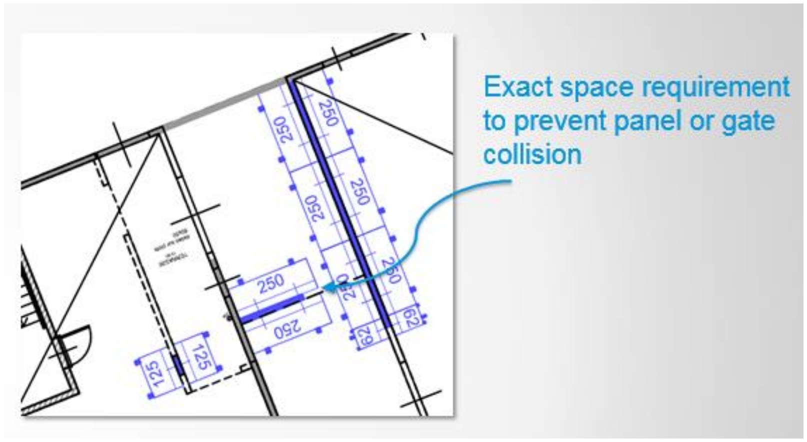
- <u>Second method: use of the array command:</u> this method happens to be simpler (less formula and visibility settings). We separate from the array the first level (because an array is a minimum of two items) and the final level (because the last level always differ from the common level). All the common level are placed in a array and Revit let us link the number of items in the array to a family parameter:



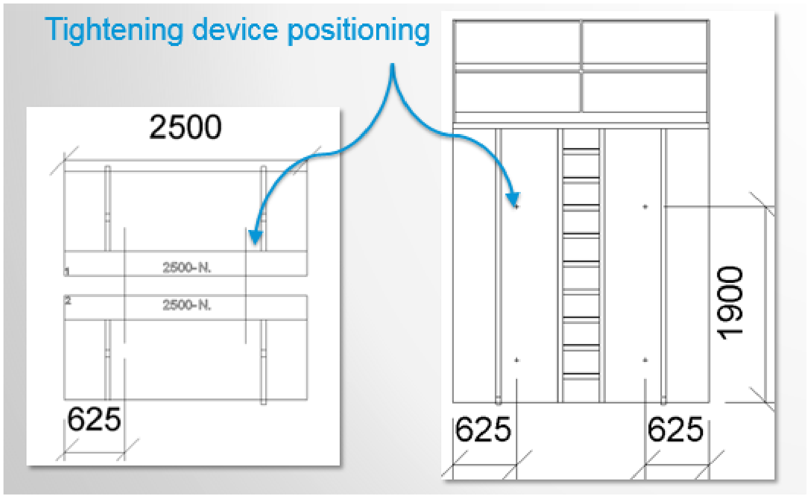
5.4 Wall formwork

The global requirements specification for this equipment library are:

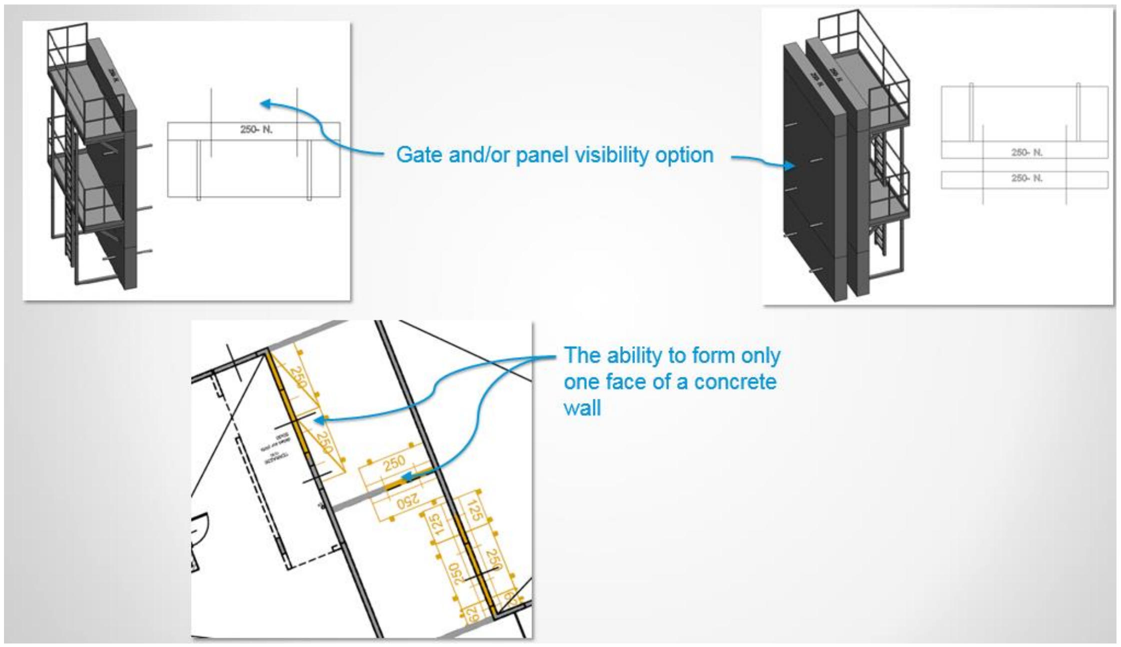
1. **The exact space requirement**
   This prevent panel or gate collision during the sequencing:



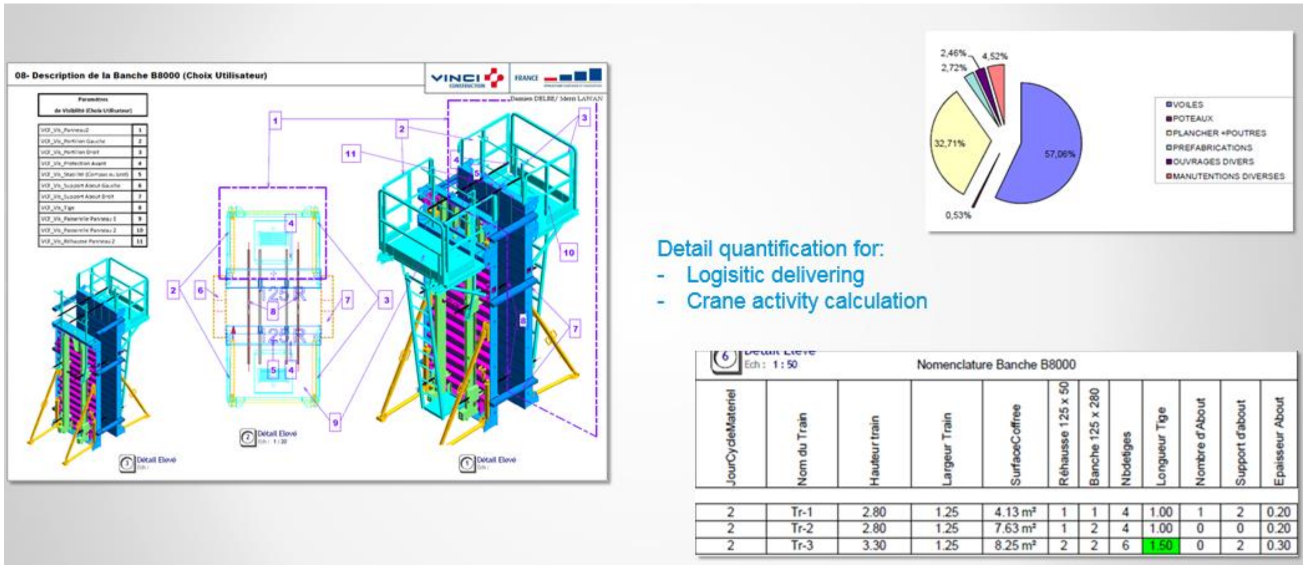
2. **The exact tightening device positioning**
   This prevent collision with concrete reinforcement for example



3. **Gate and/or panel visibility option**
   This happens when the equipment is used in space restricted conditions (like corridor) or to form only one face of a concrete wall:


Gate and/or panel visibility option
The ability to form only one face of a concrete wall
250- N.
250
125

4. **Detail scheduling day by day**
   Daily scheduling formwork accessories and global linear helps engineers calculate delivery and crane activity. In France wall formwork is mainly composed of accessories only liftable by crane that's why it represents almost 60 % of the crane activity. Example of formwork parts scheduled are:
   - Negative mold units
   - Panels, forming strip, stop-end supports
   - Number of interior and exterior angles and their length compensation
   - Panels assembly and disassembly frequency



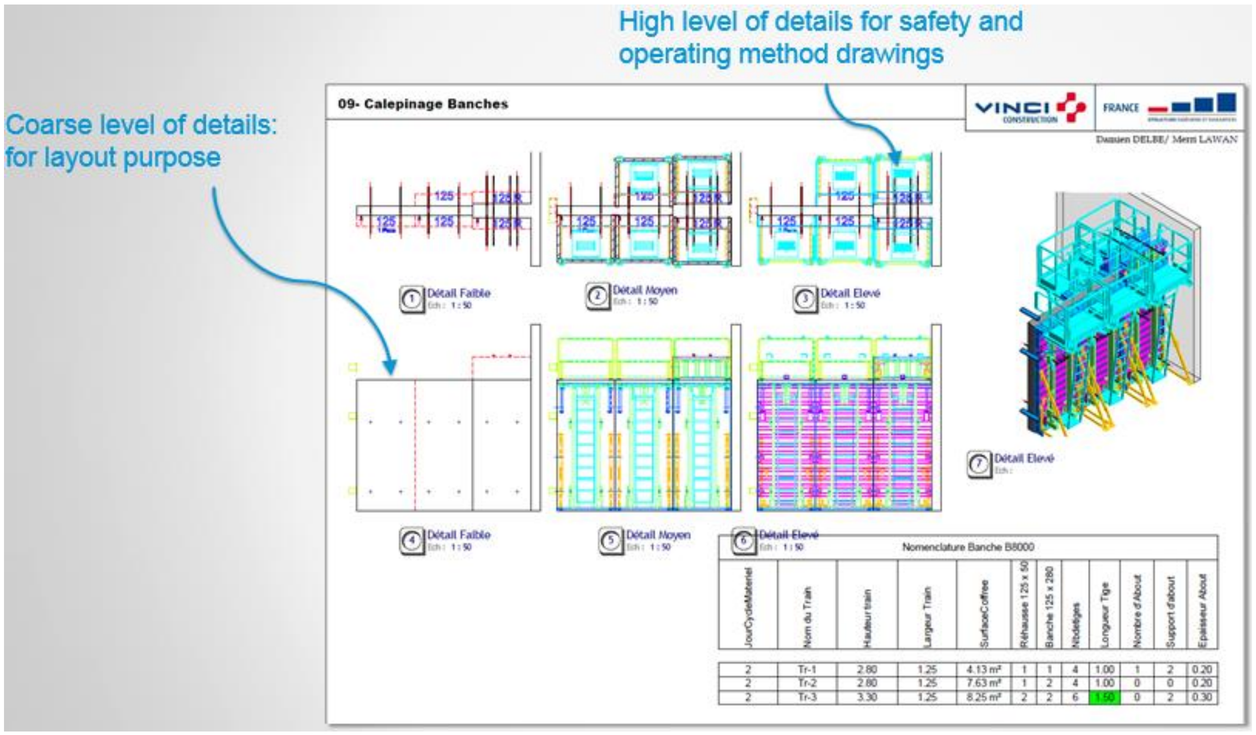
Detail quantification for:
- Logisitic delivering
- Crane activity calculation

6 Détail Elevé Ech : 1 : 50

Nomenclature Banche B8000

JourCycleMateriel	Nom du Train	Hauteur train	Largeur Train	SurfaceCoffree	Réhausse 125 x 50	Banche 125 x 280	Nbdetiges	Longueur Tige	Nombre d'About	Support d'about	Epaisseur About
2	Tr-1	2.80	1.25	4.13 m²	1	1	4	1.00	1	2	0.20
2	Tr-2	2.80	1.25	7.63 m²	1	2	4	1.00	0	0	0.20
2	Tr-3	3.30	1.25	8.25 m²	2	2	6	1.50	0	2	0.30

5. **Level of details**
   It is important to plan the visibility level of details because in a Revit Project, wall formwork family manipulation can be really time-consuming. Ideally, in the coarse visibility mode, you must keep only the relevant informations for placement if you do not want to have performance issues during the drawing process:

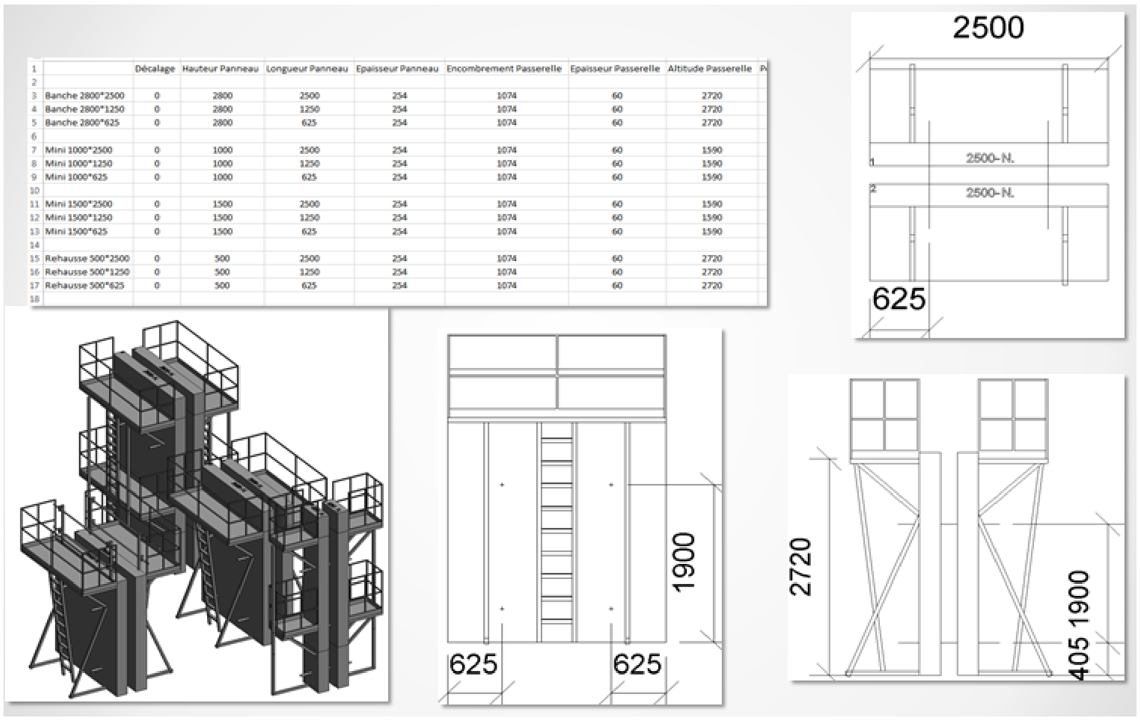

High level of details for safety and operating method drawings
Coarse level of details: for layout purpose
09- Calepinage Banches
VINCI CONSTRUCTION
FRANCE
Damien DELBE/ Merri LAWAN
1 Détail Faible Ech : 1 : 50
2 Détail Moyen Ech : 1 : 50
3 Détail Elevé Ech : 1 : 50
4 Détail Faible Ech : 1 : 50
5 Détail Moyen Ech : 1 : 50
6 Détail Elevé
7 Détail Elevé

Nomenclature Banche B8000

JourCycleMateriel	Nom du Train	Hauteur train	Largeur Train	SurfaceCoffree	Réhausse 125 x 50	Banche 125 x 280	Nbdetiges	Longueur Tige	Nombre d'About	Support d'about	Epaisseur About
2	Tr-1	2.80	1.25	4.13 m²	1	1	4	1.00	1	2	0.20
2	Tr-2	2.80	1.25	7.63 m²	1	2	4	1.00	0	0	0.20
2	Tr-3	3.30	1.25	8.25 m²	2	2	6	1.50	0	2	0.30

6. **Maintainability by creating generic components**
   Because wall formworks across different manufacturers is composed of a lots of similar modules - that can vary by height, length, thickness or position - we decided to use a generic component, i.e. a single family file (.rfa) which covers all the cases:

The data is stored in an Excel file and we can generate all the modules and superposition at once with the Revit family catalog creation feature (this also could be done using the API):



1		Décalage	Hauteur Panneau	Longueur Panneau	Epaisseur Panneau	Encombrement Passerelle	Epaisseur Passerelle	Altitude Passerelle
2								
3	Banche 2800*2500	0	2800	2500	254	1074	60	2720
4	Banche 2800*1250	0	2800	1250	254	1074	60	2720
5	Banche 2800*625	0	2800	625	254	1074	60	2720
6								
7	Mini 1000*2500	0	1000	2500	254	1074	60	1590
8	Mini 1000*1250	0	1000	1250	254	1074	60	1590
9	Mini 1000*625	0	1000	625	254	1074	60	1590
10								
11	Mini 1500*2500	0	1500	2500	254	1074	60	1590
12	Mini 1500*1250	0	1500	1250	254	1074	60	1590
13	Mini 1500*625	0	1500	625	254	1074	60	1590
14								
15	Rehausse 500*2500	0	500	2500	254	1074	60	2720
16	Rehausse 500*1250	0	500	1250	254	1074	60	2720
17	Rehausse 500*625	0	500	625	254	1074	60	2720
18								

6 What's the next step

In this chapter, I will talk about all the subjects we are just starting looking at, but we think they will significantly help furthermore our process, anticipation and productivity.

6.1 Connection to our internal purchasing division and equipment suppliers database

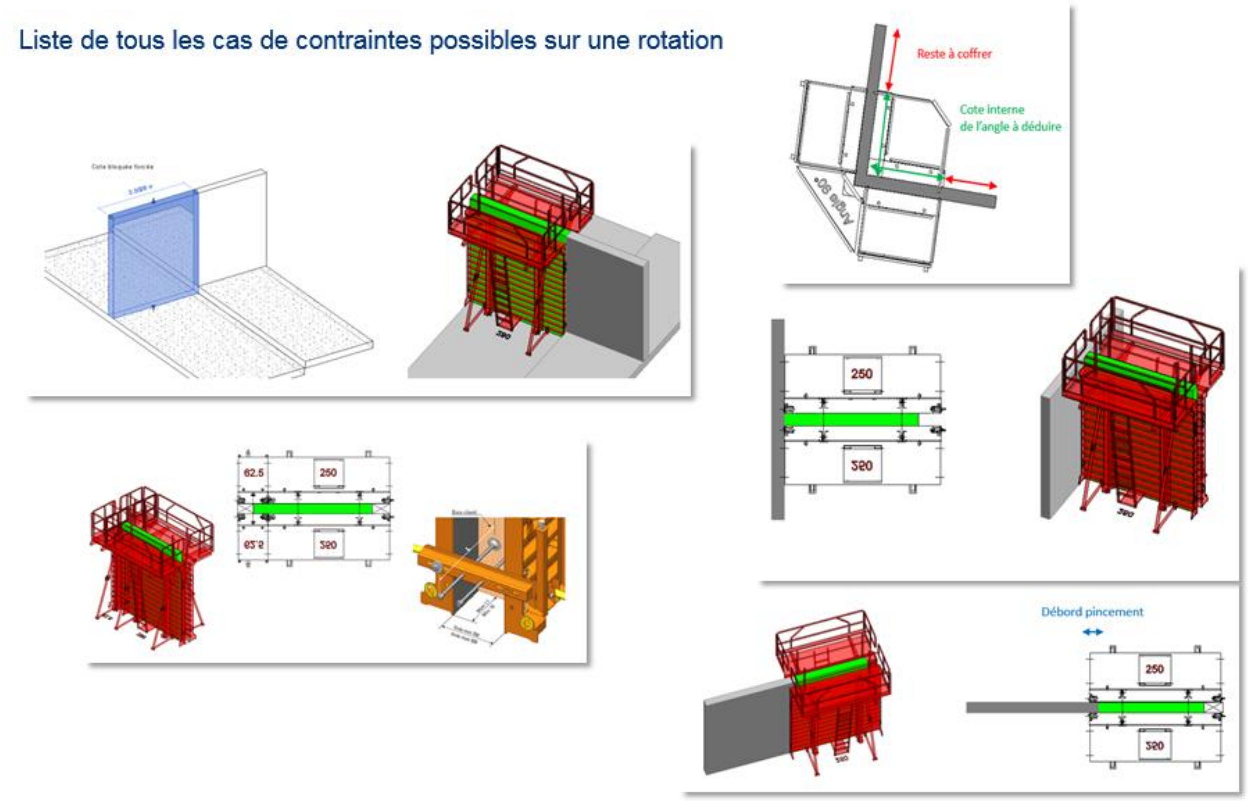
We want to migrate our classical local user Excel Database to SQL Server database in order to maintain a unique database between VINCI CONSTRUCTION FRANCE subsidiaries, our internal purchasing division and external equipment suppliers' database.

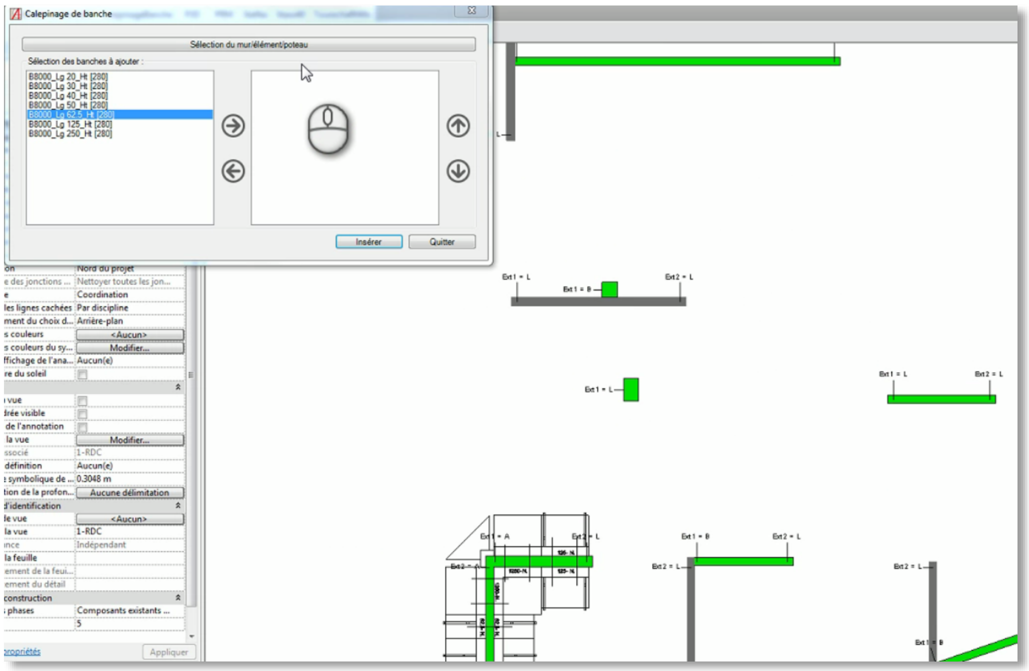
Today equipment suppliers are not using Revit but they are aware of our workflow and our needs and their design offices are really interested in the time that could be saved during the costing study. In the future we plan to work on a win-win partnership: suppliers would be responsible of the Revit library creation while following our specification.

Furthermore, our internal purchasing division deals with a lot of construction site deliveries working at the same time and spread other the territory. It could be interested for them to know in real time if the equipment will be available during a specific date range and for the specific quantity ordered, based on the building site planning and the detailed sequencing drawings.

6.2 Automatic wall formwork and shoring sheet layout creation

We have already worked on a plugin which automatically insert wall formwork module based on a daily wall sequencing drawing (made by using Revit Phase module). This plugin can read the exact length of a wall to pour picked on the drawing and suggests a list of available formwork modules that day. After the user make his choice, the plugin then insert the modules according to supplier's rules (warning messages appears if it can't explaining the reasons). Modules insertion is facilitated by reading the wall information (with, position, previous sequenced wall next to it) and automatically applying it to the formwork family (for example wall thickness value is copied into the panels spread property).




Calepinage de banche

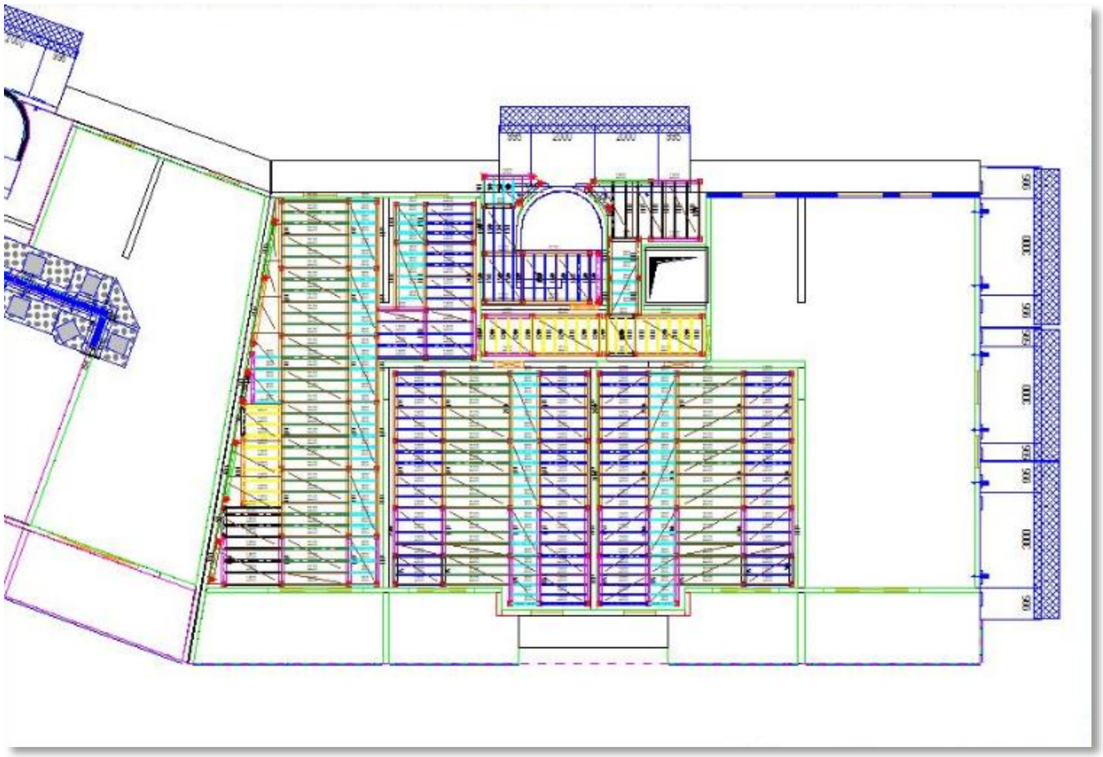
Sélection du mur/élément/poteau

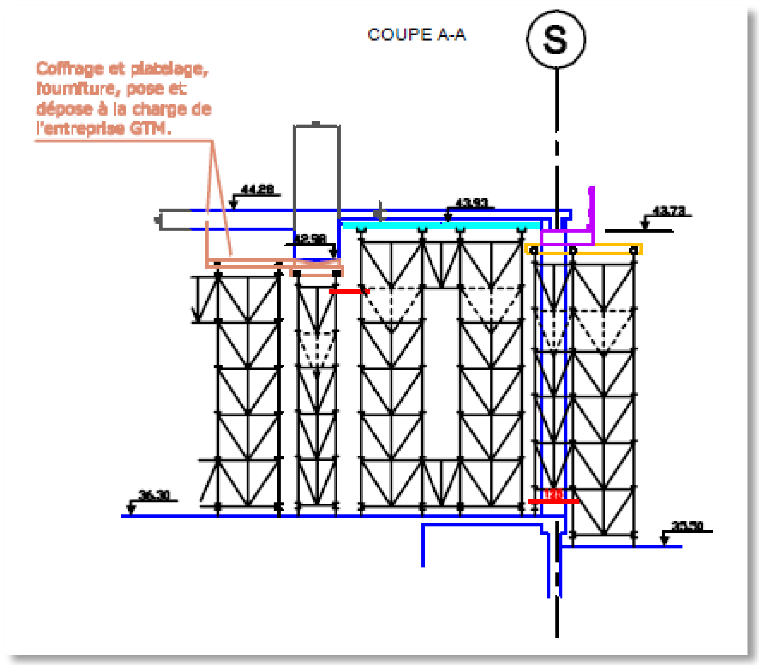
Sélection des banches à ajouter :

B8000_Lg 20_Ht [280]
B8000_Lg 30_Ht [280]
B8000_Lg 40_Ht [280]
B8000_Lg 50_Ht [280]
B8000_Lg 62.5_Ht [280]
B8000_Lg 125_Ht [280]
B8000_Lg 250_Ht [280]

on	Nord du projet
e des jonctions ...	Nettoyer toutes les jon...
e	Coordination
les lignes cachées	Par discipline
ment du choix d...	Arrière-plan
s couleurs	<Aucun>
s couleurs du sy...	Modifier...
ffichage de l'ana...	Aucun(e)
re du soleil	☐
vue	☐
drée visible	☐
de l'annotation	☐
la vue	Modifier...
ssocié	1-RDC
définition	Aucun(e)
symbolique de ...	0.3048 m
tion de la profon...	Aucune délimitation
d'identification	
le vue	<Aucun>
la vue	1-RDC
nce	Indépendant
la feuille	
ement de la feui...	
ement du détail	
construction	
phases	Composants existants ...
	5

The same idea will be applied to the shoring towers. We plan to develop a plugin which reads the slab and wall boundaries information (both vertically and horizontally) in order to automatically position scaffolds while complying with providers' requirements and Vinci's security rules.

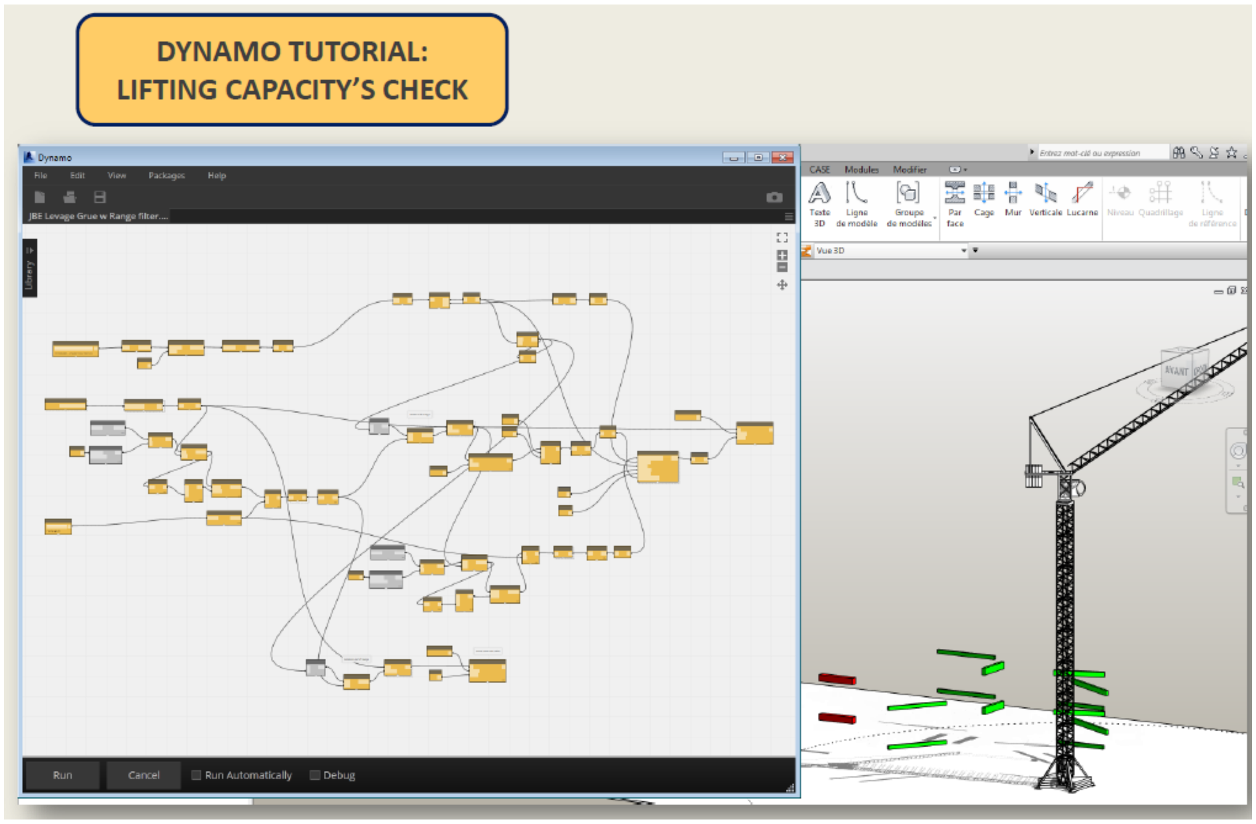



COUPE A-A
S
Coffrage et platelage, fourniture, pose et dépose à la charge de l'entreprise GTM.
44.28
43.93
43.73
42.58
36.30
35.50

6.3 Lifting crane analysis

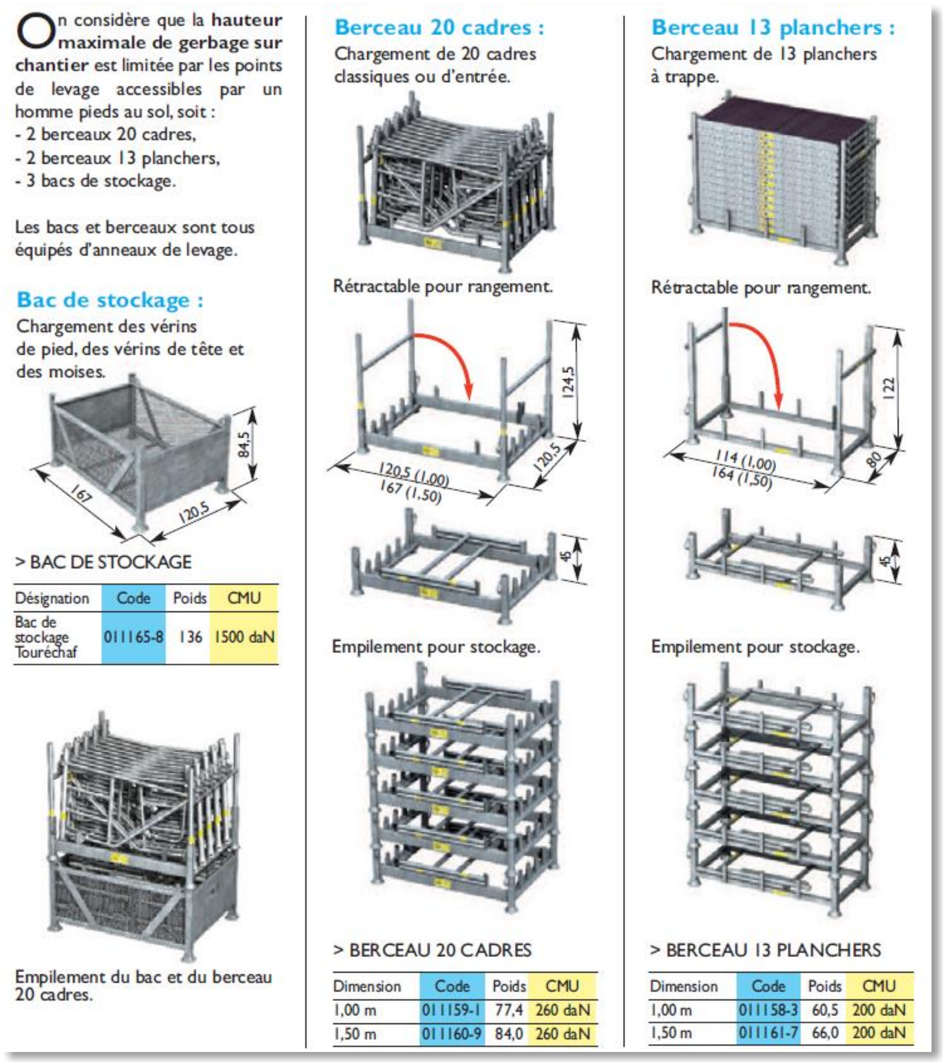
This analysis is used to be done manually in AutoCAD and with a calculator. Julien Benoit posted an interesting subject on his blog (https://aecuandme.wordpress.com/) where he explains how to make a structural framing weight analysis with Dynamo and Revit in a single click.

The idea is to read the crane family data once it is inserted (load curves, position) in order to check for each individual structural framing if it is liftable or not. This analysis helps us optimize the crane position by automatically calculating the crane activity depending on that position. It also helps us produce construction method plans with different color about whether it is precast or cast in place.



6.4 Packing and space requirement calculation

The equipment pieces ordered are delivered on the building site in trails and cradles. The suppliers packaging rules are well documented so that it is possible to automatically deduce the space requirement for the materiel storage delivery. This anticipation allows a better site organization by better plan storage area:


84,5
167
120,5
124,5
120,5 (1,00)
167 (1,50)
120,5
45
122
114 (1,00)
164 (1,50)
80
45

On considère que la **hauteur maximale de gerbage sur chantier** est limitée par les points de levage accessibles par un homme pieds au sol, soit :
- 2 berceaux 20 cadres,
- 2 berceaux 13 planchers,
- 3 bacs de stockage.

Les bacs et berceaux sont tous équipés d'anneaux de levage.

Bac de stockage :
Chargement des vérins de pied, des vérins de tête et des moises.

> BAC DE STOCKAGE

Désignation	Code	Poids	CMU
Bac de stockage Tourèchaf	011165-8	136	1500 daN

Empilement du bac et du berceau 20 cadres.

Berceau 20 cadres :
Chargement de 20 cadres classiques ou d'entrée.

Rétractable pour rangement.

Empilement pour stockage.

> BERCEAU 20 CADRES

Dimension	Code	Poids	CMU
1,00 m	011159-1	77,4	260 daN
1,50 m	011160-9	84,0	260 daN

Berceau 13 planchers :
Chargement de 13 planchers à trappe.

Rétractable pour rangement.

Empilement pour stockage.

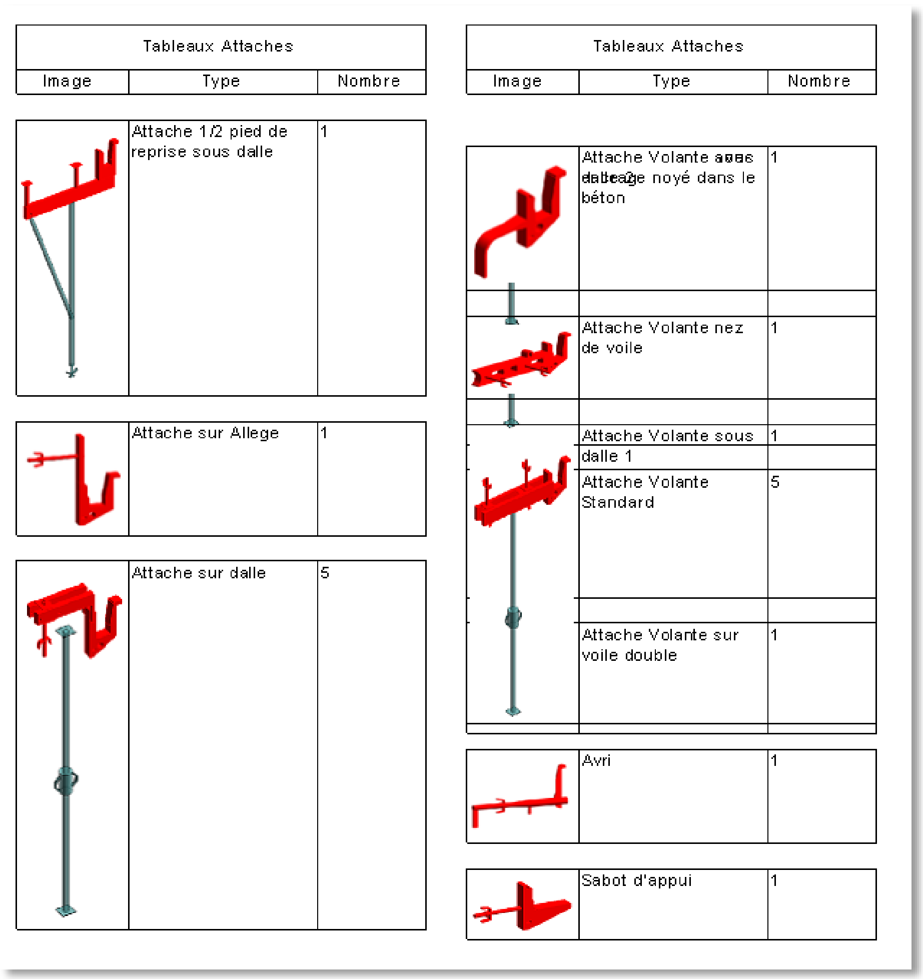
> BERCEAU 13 PLANCHERS

Dimension	Code	Poids	CMU
1,00 m	011158-3	60,5	200 daN
1,50 m	011161-7	66,0	200 daN

6.5 Equipment documentation

To produce equipment documentation, Revit provides two useful functionalities:

- The ability to embed images into an instance or a type parameter and get this image on the drawing sheets though schedules. It helps us produce more user friendly and understandable quantification tables by adding a kind of "Legend" feature:



Tableaux Attaches

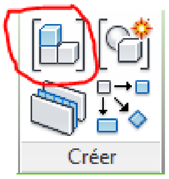
Image	Type	Nombre
	Attache 1/2 pied de reprise sous dalle	1
	Attache sur Allege	1
	Attache sur dalle	5

Tableaux Attaches

Image	Type	Nombre
	Attache Volante avec ancrage noyé dans le béton	1
	Attache Volante nez de voile	1
	Attache Volante sous dalle 1	1
	Attache Volante Standard	5
	Attache Volante sur voile double	1
	Avri	1
	Sabot d'appui	1

FIGURE 54 SCHEDULE WITH IMAGES EMBEDDED

- The assembly command feature (new since Revit 2013) helps us isolate and generate multiple views (3D, plan, section, even schedules) at once of a single Revit assembly:



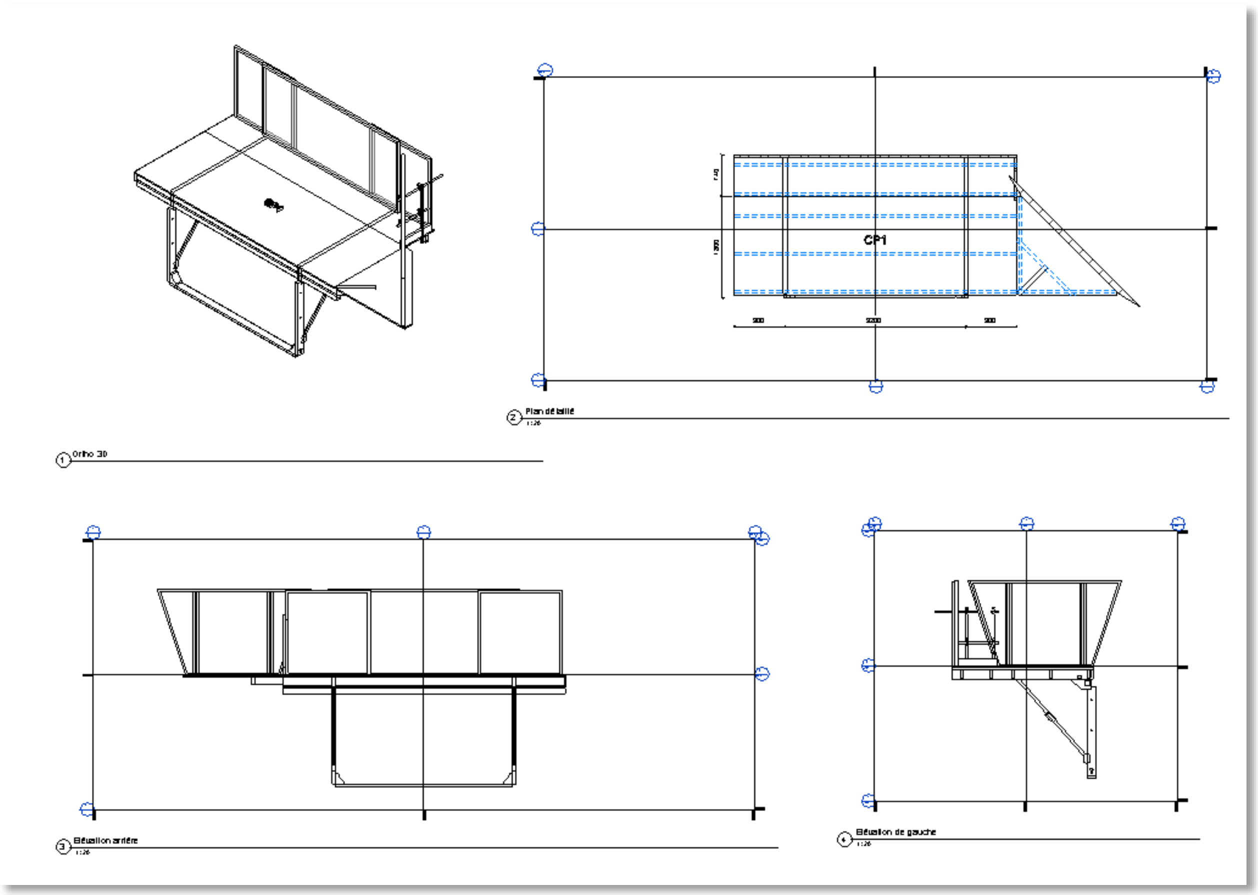
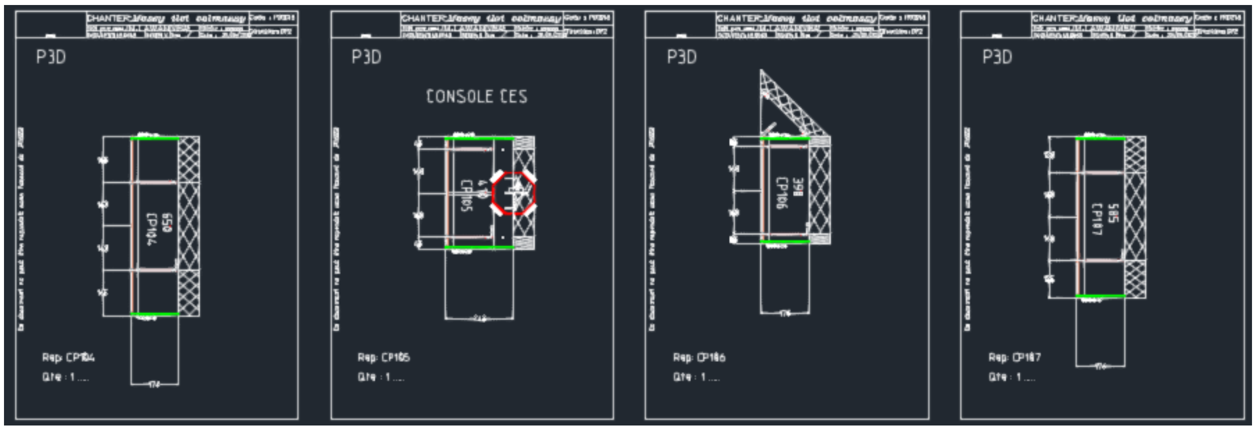


FIGURE 55 USE OF THE ASSEMBLY VIEWS TO PRODUCE EQUIPMENT DOCUMENTATION

We plan to enhance this great feature thanks to the API in order to add specific annotations automatically (like safety platform extension for example):



6.6 4D Animation for better visualization and understanding

The aim is to explain to our client or the building site workers the construction process. To do so, we are more and more testing Revit connection to 4D software visualization (like Synchro Ltd, Navisworks, 3DS max). We just started writing a document that sum up Revit library requirements in order to reuse parts of the components and animate it: for example it is required to separate moving parts before exporting or to assign specific materials).

Suite réalisation des pieux
Réalisation des poteaux préfondés (préfabriqués)
BUREAUX
Février 2015
00:28
HD